



Ektron CMS400.NET Developer Reference Manual

Version 6.0 for .NET 2.0, Rev 8

Ektron® Inc.
5 Northern Blvd., Suite 6
Amherst, NH 03031
Tel: +1 603-594-0249
Fax: +1 603-594-0258
Email: sales@ektron.com
<http://www.ektron.com>

Copyright 2006 Ektron®, Inc. All rights reserved.

EKTRON is a registered trademark of Ektron, Inc.

Version 6.0 for .NET 2.0, Rev 8, July 18, 2006

For updated versions of this manual, please visit:
<http://www.ektron.com/web-content-management-solutions.aspx?id=4020>

EKTRON, INC. SOFTWARE LICENSE AGREEMENT

**** READ THIS BEFORE LOADING SOFTWARE****

YOUR RIGHT TO USE THE PRODUCT DELIVERED IS SUBJECT TO THE TERMS AND CONDITIONS SET OUT IN THIS LICENSE AGREEMENT. USING THIS PRODUCT SIGNIFIES YOUR AGREEMENT TO THESE TERMS. IF YOU DO NOT AGREE TO THIS SOFTWARE LICENSE AGREEMENT, DO NOT DOWNLOAD.

CUSTOMER should carefully read the following terms and conditions before using the software program(s) contained herein (the "Software"). Downloading and/or using the Software or copying the Software onto CUSTOMER'S computer hard drive indicates CUSTOMER'S acceptance of these terms and conditions. If CUSTOMER does not agree with the terms of this agreement, CUSTOMER should not download.

Ektron, Inc. ("Ektron") grants, and the CUSTOMER accepts, a nontransferable and nonexplosive License to use the Software on the following terms and conditions:

1. Right to use: The Software is licensed for use only in delivered code form. Each copy of the Software is licensed for use only on a single URL. Each license is valid for the number of seats listed below (the "Basic Package"). Any use of the Software beyond the number of authorized seats contained in the Basic Package without paying additional license fees as provided herein shall cause this license to terminate. Should CUSTOMER wish to add seats beyond the seats licensed in the Basic Package, the CUSTOMER may add seats on a block basis at the then current price for additional seats (see product pages for current price). The Basic Packages are as follows:

Ektron CMS300 - Licensed for ten seats (10 named users) per URL.

Ektron CMS400.NET - Licensed for ten seats (10 named users) per URL.

Ektron eWebEditPro - Licensed for ten seats (10 named users) per URL.

Ektron eWebEditPro+XML - Licensed for ten seats (10 named users) per URL.

For purposes of this section, the term "seat" shall mean an individual user provided access to the capabilities of the Software.

The CUSTOMER may not modify, alter, reverse engineer, disassemble, or decompile the Software. This software product is licensed, not sold.

2. Duration: This License shall continue so long as CUSTOMER uses the Software in compliance with this License. Should CUSTOMER breach any of its obligations hereunder, CUSTOMER agrees to return all copies of the Software and this License upon notification and demand by Ektron.

3. Copyright: The Software (including any images, "applets," photographs, animations, video, audio, music and text incorporated into the Software) as well as any accompanying written materials (the "Documentation") is owned by Ektron or its suppliers, is protected by United States copyright laws and international treaties, and contains confidential information and trade secrets. CUSTOMER agrees to protect the confidentiality of the Software and Documentation. CUSTOMER agrees that it will not provide a copy of this Software or Documentation nor divulge any proprietary information of Ektron to any person, other than its employees, without the prior consent of Ektron; CUSTOMER shall use its best efforts to see that any user of the Software licensed hereunder complies with this license.

4. Limited Warranty: Ektron warrants solely that the medium upon which the Software is delivered will be free from defects in material and workmanship under normal, proper and intended usage for a period of three (3) months from the date of receipt. Ektron does not warrant the use of the Software will be uninterrupted or error free, nor that program errors will be corrected. This limited warranty shall not apply to any error or failure resulting from (i) machine error, (ii) Customer's failure to follow operating instructions, (iii) negligence or accident, or (iv) modifications to the Software by any person or entity other than Company. In the event of a breach of warranty, Customer's sole and exclusive remedy, is repair of all or any portion of the Software. If such remedy fails of its essential purpose, Customer's sole remedy and Ektron's maximum liability shall be a refund of the paid purchase price for the defective Products only. This limited warranty is only valid if Ektron receives written notice of breach of warranty within thirty days after the warranty period expires.

5. Limitation of Warranties and Liability: THE SOFTWARE AND DOCUMENTATION ARE SOLD "AS IS" AND WITHOUT ANY WARRANTIES AS TO THE PERFORMANCE, MERCHANTABILITY, DESIGN, OR OPERATION OF THE SOFTWARE. NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. EXCEPT AS DESCRIBED IN SECTION 4, ALL WARRANTIES EXPRESS AND IMPLIED ARE HEREBY DISCLAIMED.

NEITHER COMPANY NOR ITS SUPPLIERS SHALL BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS OR GOODWILL, LOSS OF DATA OR USE OF DATA, INTERRUPTION OF BUSINESS NOR FOR ANY OTHER INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND UNDER OR ARISING OUT OF, OR IN ANY RELATED TO THIS AGREEMENT, HOWEVER, CAUSED, WHETHER FOR BREACH OF WARRANTY, BREACH OR REPUDIATION OF CONTRACT, TORT, NEGLIGENCE, OR OTHERWISE, EVEN IF COMPANY OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS.

6. Miscellaneous: This License Agreement, the License granted hereunder, and the Software may not be assigned or in any way transferred without the prior written consent of Ektron. This Agreement and its performance and all claims arising from the relationship between the parties contemplated herein shall be governed by, construed and enforced in accordance with the laws of the State of New Hampshire without regard to conflict of laws principles thereof. The parties agree that any action brought in connection with this Agreement shall be maintained only in a court of competent subject matter jurisdiction located in the State of New Hampshire or in any court to which appeal therefrom may be taken. The parties hereby consent to the exclusive personal jurisdiction of such courts in the State of New Hampshire for all such purposes. The United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from governing this License. If any provision of this License is to be held unenforceable, such holding will not affect the validity of the other provisions hereof. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any term of this License, YOUR LICENSE IS AUTOMATICALLY TERMINATED. This License represents the entire understanding between the parties with respect to its subject matter.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, THAT YOU UNDERSTAND THIS AGREEMENT, AND UNDERSTAND THAT BY CONTINUING THE INSTALLATION OF THE SOFTWARE, BY LOADING OR RUNNING THE SOFTWARE, OR BY PLACING OR COPYING THE SOFTWARE ONTO YOUR COMPUTER HARD DRIVE, YOU AGREE TO BE BOUND BY THIS AGREEMENT'S TERMS AND CONDITIONS. YOU FURTHER AGREE THAT, EXCEPT FOR WRITTEN SEPARATE AGREEMENTS BETWEEN EKTRON AND YOU, THIS AGREEMENT IS A COMPLETE AND EXCLUSIVE STATEMENT OF THE RIGHTS AND LIABILITIES OF THE PARTIES.

Copyright 1999 - 2006 Ektron, Inc. All rights reserved.

LA10031, Revision 1.7

Securing Ektron CMS400.NET

Security Checklist

The following is a checklist of things an Administrator must do to secure **Ektron CMS400.NET**.

☐ **Change Admin user password**

☐ **Change builtin user password**

IMPORTANT!

If you changed the builtin user password during the site setup, you do not need to change it again. See ["Editing Builtin"](#) on page ii for additional information.

☐ **Remove Sample users**

IMPORTANT!

Some users in this list might not appear in your User list. This depends on which version of the software you have installed.

☐ **Admin2**

☐ **jedit**

☐ **tbrown**

☐ **jsmith**

☐ **vs**

☐ **EkExplorerUser**

☐ **Explorer**

Additional Information

You *do not* need to make these changes if you are using the CMS400.NET for demonstration or evaluation purposes. These changes should be completed once the decision is made to purchase the product and go live with your site.

IMPORTANT! Ektron recommends creating your own Administrator user and deleting the Admin user.



IMPORTANT! Ektron recommends deleting any users from the CMS400.NET that are not needed on the system.

IMPORTANT! By default, when the Ektron Explorer client is installed, it has the Explorer or EkExplorerUser username and password installed. Client side users should use their own Ektron CMS400.NET username and password when using the Ektron Explorer. To learn how to change this information, see the Administrator Manual section “Ektron Explorer” > “Configuring Ektron Explorer.”

Editing Builtin

IMPORTANT! The “builtin” user does not appear in the Users list. This user appears on the application setup screen.

To change the builtin account information, follow these steps.

1. In the Workarea, click **Settings > Configuration > Setup**.
2. Click the Edit button (.
3. Locate the **Built In User** field.
4. Edit the built in user information by changing the username and password.
5. Click the Update button (.

What's New in the 6.0 Release

- The Poll server control allows you to add a Poll or Survey to a Web form, scale it, and have it appear in an isolated display (an iFrame). See Also: ["Poll Server Control" on page 176](#)
- Eight Blog server controls have been added to give you flexibility when displaying a blog. The Blog server control displays all of the blog parts together in one server control. The other seven: BlogEntries, BlogPost, BlogRoll, BlogCategories, BlogRecentPosts, BlogRSS and BlogCalendar display each blog part individually. See Also: ["Blog Server Controls" on page 43](#)
- The AnalyticTracker server control tracks statistics about visits to your Web site. See Also: ["Content Analytics Server Control" on page 89](#)
- To collect ratings and feedback about your Web site, use the ContentRating server control. It places a scale a Web page that allows visitors to rate and comment on content on your site. See Also: ["ContentRating Server Control" on page 110](#)
- The PostResultsAt property in the FormBlock server control has been removed. This is now taken care of in the Workarea. It is located on the Post Back Message tab. See the User Manual section "Working with HTML Forms" > "Creating a New Form" > "Redirect to an Action Page and Forwarding Form Data".
- The Business Rule server control adds a Business Rule created in the CMS400.NET Workarea to a Web form. See Also: ["BusinessRules Server Control" on page 73](#)
- The Forum server control displays a Threaded Discussion Board on a Web form. See Also: ["Forum Server Control" on page 120](#)
- Ajax has been added to the Search Server Control. When you enable Ajax, site visitors to have an application like search experience. Some benefits of the Ajax Search are paging of search results and being able to place the results on a different part of the page. In addition, Ajax Searches are faster than traditional searches. See Also: ["Search Server Control" on page 182](#)
- The HTML editor server control displays the Ektron CMS400 HTML editor on a Web page. By adding the HTML editor to a Web page, site visitors can create or edit HTML content for a site or project. Below is an example of the editor with the default toolbars. See Also: ["HTML editor Server Control" on page 122](#)
- The ContentList server control displays a list of content blocks created from a comma delimited list. See Also: ["ContentList Server Control" on page 100](#)
- The FolderBreadcrumb server control displays a sitemap style breadcrumb trail on your Web pages. See Also: ["FolderBreadcrumb Server Control" on page 69](#)

-
- The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. See Also: "[DesignTimeDiagnostic Server Control](#)" on page 115
 - The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. See Also: "[SiteMap Server Control](#)" on page 235
 - Added list of additional developers resources. See Also: "[List of Developer Resources](#)" on page 2

Summary Table of Contents

Securing Ektron CMS400.NET	i
What's New in the 6.0 Release	iii
About This Manual	1
List of Developer Resources	2
Introduction to Ektron CMS400.NET Server Controls	7
Blog Server Controls	43
BreadCrumb Server Control	59
FolderBreadcrumb Server Control	69
BusinessRules Server Control	73
Calendar Server Control	75
Collection Server Control	82
Content Analytics Server Control	89
ContentBlock Server Control	91
ContentList Server Control	100
ContentRating Server Control	110
DesignTimeDiagnostic Server Control	115
FormBlock Server Control	117
Forum Server Control	120
HTMLEditor Server Control	122
LanguageAPI Server Control	130
LanguageSelect Server Control	133
ListSummary Server Control	135
Login Server Control	143
Membership Server Control	146
Menu Server Control	152
DhtmlMenu Server Control	160
MetaData Server Control	167
MetaDataList Server Control	171
Poll Server Control	176
RssAggregator Server Control	179
Search Server Control	182

IndexSearch Server Control.....	218
SiteMap Server Control	235
 Dreamweaver Extension Support	238
 Standard ASP.NET Methods	403
 Custom ASP Functions	411
Login Function	414
Content Block Function.....	416
Add Content.....	424
Search Functions	428
Search Display Function	438
List Summary Function	442
XML List Summary Function.....	450
Archive List Summary Function	455
Single Summary Function.....	456
RSS Summary Function	460
Collection Function.....	465
RSS Collection Function	469
XML Collection Function	471
Random Content	476
Random Summary	477
Event Calendar Function.....	478
Calendar Function.....	481
Form Function	485
MultiPurpose Function	487
Metadata Function.....	488
Show Site Language Function.....	494
DHTML Menus	495
XML Menu	499
 Custom ColdFusion Functions	501
 Custom PHP Functions	521
Custom API.....	542
 Custom JSP Functions	543

Web Services	561
Syndicating Web Content as RSS	580
Syndicating via .NET.....	581
Learning About Visual Studio	583
Customizing Validation.....	584
Programmatic Search API	602
Cascading Style Sheets	604
Additional Information	647

Detailed Table of Contents

Securing Ektron CMS400.NET	i
Security Checklist.....	i
Additional Information	ii
Editing Built-in.....	ii
 What's New in the 6.0 Release	iii
 About This Manual	1
 List of Developer Resources.....	2
Ektron's DevCenter.....	2
User Discussion Boards	2
Ektron Blogs	2
Ektron FileShare.....	3
Tutorials	3
External Resources	3
Knowledge Base Articles.....	3
Ektron's Developer Training Course.....	3
Developer Site Provided with Ektron CMS400.NET Install	3
Ektron CMS400.NET API Documentation	4
API Help Inside Visual Studio 2005	4
API Help in PDF Format.....	5
From the Desktop	5
From the Workarea	5
From www.ektron.com	5
Ektron Support and Product Pages	5
Supporting Documents	6
 Introduction to Ektron CMS400.NET Server Controls	7
Opening the Sample Project.....	7
Setting Up a Template	9
Making the Server Controls Available	10
Removing the Server Controls	12
Updating the Server Controls	12
Inserting Server Controls Using Drag and Drop.....	12
Adding a Server Control to a Template.....	12
ServerControlWS.aspx.....	14

Using CMS Explorer to Browse Your Ektron CMS400.NET Site...	15
Build the Project	16
Inserting Server Controls Programmatically.....	16
Steps to Insert a Server Control Programmatically.....	17
Step 1: Declare the Namespace	17
Step 2: Create an Instance of the New Control	18
Step 3: Set the Properties	18
Step 4: Call the Fill Method	19
Step 5: Use .text Property to Display the Control.....	19
Server Control Properties in Code Behind	20
Displaying Properties for a Control	20
Mutual Server Control Properties in Code Behind	21
Accessing Additional Properties.....	23
Accessing Items in an Array.....	28
Referencing the Page Property	28
Customizing Server Controls.....	29
Recognizing the Server Control Within the HTML	30
Recognizing the Server Control Within the Code Behind	30
Customizing the Server Control in the Code Behind.....	30
Troubleshooting Error Creating Control Message.....	31
Using Drag & Drop and Programmatically Together	31
Data Binding with Server Controls.....	33
Steps to Data Binding Using Drag and Drop Server Controls	34
Caching with Server Controls	36
Caching Individual Content.....	37
Page Level Caching	38
List of Server Controls	39
Blog Server Controls	43
Using the Blog Server Control vs. Using the Individual Blog	
Server Controls.....	43
Blog Server Control.....	43
Blog Server Control Properties.....	44
BlogEntries Server Control.....	45
BlogEntries Server Control Properties.....	46
BlogPost Server Control	48
BlogPost Server Control Properties.....	48
BlogCalendar Server Control	50
BlogCalendar Server Control Properties	50
Using the BlogCalendar and BlogEntries Server Controls Together	51
Steps to Use BlogCalendar and BlogEntries Together.....	51
BlogRoll Server Control	54
BlogRoll Server Control Properties.....	54
BlogCategories Server Control	55
BlogCategories Server Control Properties	55
BlogRecentPosts Server Control	56

BlogRecentPosts Server Control Properties	56
BlogRSS Server Control	57
BlogRSS Server Control Properties	58
BreadCrumb Server Control	59
BreadCrumb Properties Apply to Web Forms, Not Web Pages	61
Types of BreadCrumb Server Control Properties.....	61
Properties that Determine a Breadcrumb Trail's Appearance	61
Properties that Determine How Pages Associated with the Form	
Appear on a Breadcrumb Trail.....	62
BreadCrumb Server Control Properties	62
BreadCrumb Metadata Type	65
Using the BreadCrumb Server Control.....	66
Making a Content Block's Title Appear in the Breadcrumb Trail	67
Steps to Use a Content Block's Title as a Breadcrumb	67
FolderBreadcrumb Server Control.....	69
BreadCrumb Server Control Properties	69
Using the BreadCrumb Server Control.....	71
BusinessRules Server Control	73
BusinessRules Server Control Properties	73
Calendar Server Control.....	75
Calendar Server Control Properties.....	76
Calendar Style Sheet	80
Collection Server Control.....	82
Collection Server Control Properties.....	82
Example of ecmNavigation Display	85
ecmNavigation XSL code.....	85
Example of ecmTeaser Display	85
ecmTeaser XSL code.....	86
Retrieving the XML Structure of a Collection.....	86
Using the Collection Server Control Programmatically Exam-	87
ple.....	87
Content Analytics Server Control.....	89
Content Analytics Server Control Properties.....	89
Ektron Properties.....	89
ContentBlock Server Control.....	91
Static Content Block.....	91
ContentBlock Server Control Properties.....	91
Dynamic Content Block	93
XML Content Block.....	95
The OverrideXslt and DisplayXslt Properties	96
Retrieving the XML Structure of an XML Content Block.....	98
Using the ContentBlock Server Control Programmatically Ex-	98
ample	98

ContentList Server Control	100
ContentList Server Control Properties	100
Using the DefaultContentID and MetaTag Properties.....	105
Using the ContentID Property to Display a Content List.....	109
ContentRating Server Control	110
Content Rating Server Control Properties	110
Ektron Properties.....	110
Ektron Label Properties	112
Initial Graph Display	112
Initial Graph Properties	113
Complete Graph Display	114
Complete Graph Properties	114
DesignTimeDiagnostic Server Control	115
DesignTimeDiagnostic Server Control Properties	116
FormBlock Server Control	117
FormBlock Server Control Properties	118
Automatic versus Manual Generation of Form Tags.....	119
Forum Server Control	120
Server Control Properties	120
Ektron Properties.....	120
Updating the Page Command	121
HTMLEditor Server Control.....	122
Introduction.....	122
Working with Ektron CMS400.NET Content	123
Working with Non Ektron CMS400.NET Content	124
Uploading Files with an Action Page.....	124
HTMLEditor Server Control Properties.....	124
ShowCMSToolbar	127
ToolbarLevel	128
LanguageAPI Server Control	130
LanguageSelect Server Control vs. LanguageAPI Server Control	130
LanguageAPI Server Control Properties	130
Code Behind Only Properties.....	131
Using the LanguageAPI Server Control Programmatically .	132
LanguageSelect Server Control.....	133
LanguageSelect Server Control vs. LanguageAPI Server Control	133
LanguageSelect Server Control Properties.....	133
ListSummary Server Control	135
ListSummary Server Control Properties	135
OrderKey Property.....	140
List Summary Example	141
Retrieving the XML Structure of a List Summary	142

Login Server Control	143
Placing a Login Button.....	143
Login Server Control Properties	144
Membership Server Control	146
Membership Server Control Properties	147
DisplayMode.....	150
Menu Server Control.....	152
Menu Server Control vs. DhtmlMenu Server Control	152
Menu Server Control Properties.....	153
Using DisplayXslt Samples.....	154
SampleMenu	154
SampleMenu XSL code	156
TreeMenu	156
TreeMenu XSL code	158
Retrieving the XML Structure of a Menu.....	158
DhtmlMenu Server Control.....	160
Menu Server Control vs. DhtmlMenu Server Control	161
DhtmlMenu Server Control Properties.....	161
Using a Style Sheet	165
Formatting the DhtmlMenu Server Control	166
MetaData Server Control	167
MetaData Server Control vs. MetaDataList Server Control ..	167
MetaData Server Control Properties	167
Using the Simple Dublin Core Metadata Standard	168
Using the MetaData Server Control.....	169
MetaDataList Server Control	171
MetaData Server Control vs. MetaDataList Server Control ..	171
MetaDataList Server Control Properties	171
Retrieving the XML Structure of a MetadataList	175
Poll Server Control.....	176
Poll Server Control Properties	176
Automatic versus Manual Generation of Form Tags.....	178
RssAggregator Server Control	179
RSSAggregator Server Control Properties	179
Using the RssAggregator Server Control.....	181
Search Server Control	182
Introduction.....	182
Search Server Control vs. IndexSearch Server Control.....	183
Search Server Control Properties	183
Property Usage Table	184
Property Descriptions	185
Ajax Search	197
ResultTagId	198
Text Box Examples	199

Button Image Source Examples	199
Search Button Text Examples	200
Display Examples	201
ShowDate Examples	202
ShowExtendedSearch	202
TargetFolderDescriptionList	205
Using the Advanced Search Properties.....	206
Basic and Standard Searches	208
Advanced Search	209
Using Standard and Advanced Searches Together	209
Fieldset Properties	212
Retrieving the XML Structure of a Search	213
Displaying Search Results on a Separate Page.....	214
Using a Style Sheet	216
Formatting the Search Server Control.....	217
IndexSearch Server Control.....	218
IndexSearch Server Control vs. Search Server Control.....	218
IndexSearch Server Control Properties.....	218
SearchParmXML Property.....	225
Using the IndexSearch Server Control Programmatically Ex-	
ample	228
Programmatically Predefined General Search Results Re-	
placed By Specific Search Results	231
Turn Off Caching In Netscape Navigator and Mozilla Firefox....	234
SiteMap Server Control	235
Sitemap Server Control Properties	235
Using the Sitemap Server Control	237
 Dreamweaver Extension Support	238
Requirements	238
Preparing Dreamweaver for Ektron CMS400.NET Functions	238
Installing the Extensions.....	239
Setup	243
Editing and Saving Ektron CMS400.NET Content in Dream-	
weaver.....	247
Accessing Manage Content Commands... ..	247
Edit Content.....	248
Save Content.....	251
Saving New Content	251
Saving Existing Ektron CMS400.NET Content	254
Using the Ektron CMS400.NET Extensions.....	255
Custom Function Symbols.....	256
Ektron CMS400.NET Command Menus.....	260

Inserting Custom Functions	266
CMS Function.....	266
Analytics	266
Blog	269
Blog Calendar.....	272
Blog Categories	275
Blog Entries	278
Blog Post	281
Blog Recent Posts	284
Blog Roll	287
Blog RSS	290
BreadCrumb	293
Calendar	299
Collection.....	302
Content Block	306
Content List	310
Content Rating.....	315
Content XSLT Tag... ..	317
DHTML Menu	319
Discussion Forum.....	326
Display By Meta Value.....	328
Dynamic Content Block	333
Folder Bread Crumb	336
Form	338
Language API.....	342
Language Select Box	344
List Summary.....	346
Login.....	352
Membership Control	355
Metadata for ASP.NET Pages	359
Metadata Function for ASP, CF, JSP or PHP Pages	362
Multipurpose Content Block.....	365
Poll	367
Random Content	370
Random Summary.....	371
RSS Aggregator	373
Search	375
Search Display	387
Single Summary	390
Site Map	392
Standard Menu	394
XML Index Search	397
 Standard ASP.NET Methods	 403
Replacement of Standard Methods	403
Referencing Methods in Visual Basic	404
Adding a New Web Form.....	404

Requirements for Standard Methods	405
Standard Methods.....	406
Addcontent	407
Add Content Example	409
Custom ASP Functions	411
Login Function	414
Login Tag Example	414
Placing Your Login Button.....	415
Content Block Function.....	416
Content Block Examples.....	418
Static Content Block	419
Dynamic Content Block	419
XML Content Block.....	421
The Function's Arguments	421
Dynamic and Static Combination.....	423
Add Content.....	424
Add Content Example	425
Search Functions	428
Search Tag Example	430
Search Tag Attributes.....	431
Starting Folder	431
Recursive.....	431
Target Page	431
Text Box.....	432
Maximum Characters	432
Button Image Source.....	433
Button Text.....	434
Font Face	434
Font Size	436
Horizontal	436
Spare 1	437
Search Display Function	438
Search Display Example	438
Search Display with Default Content	439
Search Display Attributes	439
Max Number of Content Blocks Returned	440
Style Info.....	441
Show Date	441
List Summary Function	442
List Summary Example	444
List Summary Attributes	445
Starting Folder	445
Recursive.....	445
Show Summary	446

Style Info.....	446
Order By	447
Ascending/Descending	447
Show Info.....	448
Max Number	449
XML List Summary Function.....	450
Archive List Summary Function	455
Single Summary Function.....	456
Single Summary Example	456
Single Summary Attributes.....	457
Content ID	457
Show Summary	457
Style Info.....	458
Show Info.....	458
Spare 1	459
RSS Summary Function	460
RSS Summary Example	460
RSS Summary Attributes	463
Starting Folder	463
Recursive.....	463
Order By	463
Ascending/Descending	464
Max Number	464
Collection Function.....	465
Collection Examples.....	465
The ecmNavigation Display Function	465
The ecmTeaser Display Function	466
Collection Attributes	468
ID	468
DisplayFunction	468
RSS Collection Function	469
XML Collection Function	471
Random Content	476
Random Summary	477
Event Calendar Function.....	478
Event Calendar Style Sheet	478
Event Calendar Sample.....	478
Calendar Function.....	481
Calendar Style Sheet	483
Event Calendar Sample.....	483
Form Function	485
MultiPurpose Function	487
Metadata Function.....	488
Creating an Item List	488

ecmMetadata Example	489
More ecmMetadata Examples	490
Using a Dynamic Metadata Tag	491
ecmMetadata Attributes	491
Item List	491
Spare 1	493
Show Site Language Function	494
DHTML Menus	495
Additional Documentation in Ektron Knowledge Base	495
Displaying a Menu on a Web Page	495
GenerateMenu and DropMenu	495
Displaying Menus in a Multi-language System	496
Customizing the Menu's Appearance	496
ek_Menu.js	496
FontFamily	498
menuLiteBgColor	498
XML Menu	499

Custom ColdFusion Functions501

Login Function	503
Login Tag Example	503
Content Block Function	503
Static Content Block	504
Dynamic Content Block	504
XML Content Block	504
Add Content	504
Add Content Example	505
Random Content Function	505
Random Summary Function	506
Search Function	506
Search Tag Example	506
Search Display Function	507
Search Display Example	507
List Summary Function	507
List Summary Example	507
XML List Summary Function	508
XML List Summary Example	508
RSS List Summary Function	508
List Summary Example	508
Archive List Summary Function	509
Single Summary Function	509
Single Summary Example	509
Collection Function	509
Example Using the ecmNavigation Display Function	510
Example Using the ecmTeaser Display Function	511

Collection Variables	512
DisplayFunction	512
XML Collection Function.....	513
RSS Collection Function.....	513
Event Calendar Function	513
Event Calendar Sample.....	516
Form Function.....	516
MultiPurpose Function	516
Metadata Function	517
DHTML Menus.....	518
Additional Documentation in Ektron Knowledge Base	519
Displaying a Menu on a Web Page	519
GenerateMenu and DropMenu	519
Customizing the Menu's Appearance.....	519
ek_Menu.js.....	520
XML Menu Function.....	520
Language Select Function	520
 Custom PHP Functions	 521
Login Function.....	524
Login Function Example	524
Content Block Function	524
Static Content Block	525
Dynamic Content Block	525
XML Content Block.....	525
Add Content	525
Add Content Example.....	526
Search Function.....	526
Search Function Example	527
Search Display Function.....	527
Search Display Function Example.....	527
List Summary Function.....	527
List Summary Example.....	528
Archive List Summary Function.....	528
Single Summary Function	528
Single Summary Example	528
XML List Summary Function	529
RSS List Summary Function	529
Collection Function	529
Collection Examples	530
Example Using the ecmNavigation Display Function.....	530
Example Using the ecmTeaser Display Function.....	531
Collection Variables	532
DisplayFunction	532
XML Collection Function.....	532

RSS Collection Function	533
Random Content Function	533
Random Summary Function	533
Event Calendar Function	534
Event Calendar Sample.....	536
Form Function.....	536
MultiPurpose Function	537
Metadata Function	537
DHTML Menus	539
Additional Documentation in Ektron Knowledge Base	540
Displaying a Menu on a Web Page	540
GenerateMenu and DropMenu	540
Customizing the Menu's Appearance	540
ek_Menu.js.....	540
XML Menu Function.....	541
Language Select Function	541
Custom API.....	542

Custom JSP Functions543

Calendar Function	545
Calendar Sample.....	546
Event Calendar Function	546
Event Calendar Sample.....	546
Collection Function	547
Collection Sample.....	547
XML Collection Function.....	547
Random Content Function.....	548
Random Summary Function.....	548
RSS Collection Function	548
Add Content	549
Add Content Example.....	549
Content Block Function	549
Static Content Block	550
Dynamic Content Block	550
XML Content Block.....	550
Form Function.....	550
Language Select Function	552
Login Function	552
Login Function Example	553
DHTML Menus	553
Additional Documentation in Ektron Knowledge Base	553
Displaying a Menu on a Web Page	553
GenerateMenu and DropMenu	553
Customizing the Menu's Appearance	554
ek_Menu.js.....	554

XML Menu Function.....	554
Metadata Function	555
Search Function.....	557
Search Function Example	558
Search Display Function.....	558
Search Display Function Example.....	558
List Summary Function.....	559
List Summary Example.....	559
XML List Summary Function	559
RSS List Summary Function	559
Single Summary Function	560
Single Summary Example	560
 Web Services	 561
Introduction to Web Services	561
CMS400.NET's Implementation of Web Services	561
Benefits of Using Web Services	561
WS Components and CMS400.NET Architecture	562
Software and Hardware Requirements	562
Installation of Files	562
Setup Instructions	562
Providing Web Service Information in Your CMS	564
CMS400.NET Methods.....	564
The CMS400.NET Content Sample Page	567
The Search Display Method Parameters	570
Sample Output from One Method.....	572
GetContentBlock.....	572
The News.aspx Sample (non-XML).....	573
Identifying the Code Behind File	573
Setting Up an Instance of Web Services.....	573
Getting Content Blocks	573
The News.aspx Sample (XML).....	575
Setting Up Instances of Web Services.....	575
Getting Content Blocks	575
Creating a Page that Provides a Web Service.....	579
 Syndicating Web Content as RSS	 580
What is RSS Syndication?	580
The RSS Specification.....	580
RSS Syndication in Ektron CMS400.NET	580
Syndicating via .NET.....	581
Architecture of RSS Syndication	581
aspx pages	581
Retrieving a Collection or List Summary as an RSS Feed.....	581
Creating Your Own aspx Collection Page	581

Example.....	581
Remarks.....	581
Example.....	582
Creating Your Own aspx List Summary Page	582
Example.....	582
Remarks.....	582
Example.....	582
Learning About Visual Studio	583
Grid Layout vs. Flow Layout.....	583
Customizing Validation.....	584
Customizing Validation Options.....	584
Validation Elements in the Configuration Data	584
Validation Attributes.....	585
Validation Sub-elements	585
Choice Sub-element	585
Choice Attributes.....	586
Choice Sub-Elements	587
Defining a Schema Fragment	587
Defining a Calculation	588
Defining Validation	589
Defining an Error Message	590
Defining Custom Validation	590
Saving Invalid Documents	594
Custom Script that Handles Saving Invalid Files	594
The onerror Event	595
Calculated Fields	595
Icons on the Select Field or Group Screen.....	595
Learn More about XPath	597
XPath Operators.....	597
XPath Functions	598
XPath References.....	601
Programmatic Search API	602
Guidelines for Creating a Search	602
Cascading Style Sheets.....	604
reportchart.css.....	604
blogs.css.....	608
Layout Styles.....	608
Header Styles.....	609
Entry Styles	610
Calendar Styles.....	610
Blogroll Styles	613

Blog Categories Styles	614
Blog Recent Posts Styles	615
calendarStyles.css	616
Single Day View (Day)	617
Month View (month)	620
Inline View (inline)	632
Month-Up with Event (monthupwithevent)	634
Event Type Control (used in month display)	641
designformentry.css	644
pop_style.css	644
CMS400Demo.css	645
Developer Styles.css	645
Default.css	645
List Summary OrderedList.css	645
List Summary unorderedlist.css	645
Collection OrderedList.css	645
Collection unorderedlist.css	645
CSSMenu.css	645
Search unorderedList.css	645
Discussion Board CSS Files	645

Additional Information	647
Supporting Documents	647
Support	647
Sales	647
Information	647

About This Manual

This manual can be used as a reference for the developer, or the person who is setting up your Ektron CMS400.NET Web site.

To utilize the full potential of an Ektron CMS400.NET driven Web site, learning and using the server controls supplied by Ektron is essential. These server controls help make your site more robust and easier to maintain. For more information, see ["Introduction to Ektron CMS400.NET Server Controls" on page 7](#).

This document explains how to manually use custom tags for each platform supported by Ektron CMS400.NET:

- ["Standard ASP.NET Methods" on page 403](#)
- ["Custom ASP Functions" on page 411](#)
- ["Custom ColdFusion Functions" on page 501](#)
- ["Custom PHP Functions" on page 521](#)

Besides explaining how to insert the functions to your Web templates manually, this manual also explains how to create Ektron CMS400.NET templates in a WYSIWYG environment by using the Ektron supplied server controls for Visual Studio 2005 or by using the Dreamweaver extension supplied with the installation. See ["Dreamweaver Extension Support" on page 238](#) for additional information.

Ektron CMS400.NET has also implemented support for Web Services as well as syndication using RSS. Information about configuration and use of these features is also found in the following sections:

- ["Web Services" on page 561](#)
- ["Syndicating Web Content as RSS" on page 580](#)

List of Developer Resources

In addition to the information in this manual, the following resources provide information on developing your site. Below is a quick reference list of resources.

- "Ektron's DevCenter" on page 2
 - <http://dev.ektron.com>
- "Ektron's Developer Training Course" on page 3
 - <http://www.ektron.com/cms400-web-cms.aspx?id=2653>
- "Developer Site Provided with Ektron CMS400.NET Install" on page 3
 - <http://localhost/CMS400Demo/developer/default.aspx>
- "Ektron CMS400.NET API Documentation" on page 4
 - <http://www.ektron.com/web-content-management-solutions.aspx?id=4020>
- "Ektron Support and Product Pages" on page 5
 - http://www.ektron.com/support/cms400_support.aspx
 - <http://www.ektron.com/web-content-management-solutions.aspx>
- "Supporting Documents" on page 6
 - <http://www.ektron.com/web-content-management-solutions.aspx?id=4020>

Ektron's DevCenter

- <http://dev.ektron.com>

The Ektron DevCenter includes a wide array of resources to assist our developers and partners. Whether you're looking for a solution to a specific problem or just seeing what you might be able to do with CMS400.NET, you should be able to find what you need here.

User Discussion Boards

- <http://dev.ektron.com/forum.aspx>

Use the Forum to exchange ideas and information about Ektron, Inc. products.

Ektron Blogs

- <http://dev.ektron.com/blogs.aspx>

With Blogs, learn what developers and engineers at Ektron, Inc. are thinking.

Ektron FileShare

- <http://dev.ektron.com/fileshare.aspx>

The Ektron FileShare is a system designed to allow developers to share server controls, code snippets, classes and other tools that may be useful to other developers.

Tutorials

- <http://dev.ektron.com/tutorials.aspx>

This area provides video tutorials to help you learn how to use CMS400.

External Resources

- <http://dev.ektron.com/template.aspx?id=2756>

The sites listed on this page contain tutorials, developer news, or other materials that might be useful to you while developing your CMS400.NET site. These sites are not run by Ektron, Inc.

Knowledge Base Articles

- http://dev.ektron.com/kb_search.aspx

These articles provide information about all Ektron, Inc. products.

Ektron's Developer Training Course

- <http://www.ektron.com/cms400-web-cms.aspx?id=2653>

Why should you become a Certified Ektron Developer?

- Learn best practices for deploying an Ektron CMS400.NET driven Web site
- Learn how to extend CMS400.NET to do more of what you want
- See real-world examples of how an Ektron CMS is deployed
- Be an expert on the most commonly deployed CMS
- Add "Certified Ektron Developer" to your resume

Developer Site Provided with Ektron CMS400.NET Install

- <http://localhost/CMS400Demo/developer/default.aspx>

NOTE [Replace localhost with location of your site.](#)

Ektron has compiled a list of several different examples of developing your Web site with Ektron CMS400.NET. These examples show how easy it is to

create robust Web pages by using "out-of-the-box" functionality including server controls.

Ektron CMS400.NET API Documentation

The API documentation provides a way to learn about the methods and functions that are available when customizing your Ektron CMS400.NET site. The following APIs are documented:

- Server Control API
- Business API

Access the API documentation from within Visual Studio 2005 or in PDF format.

API Help Inside Visual Studio 2005

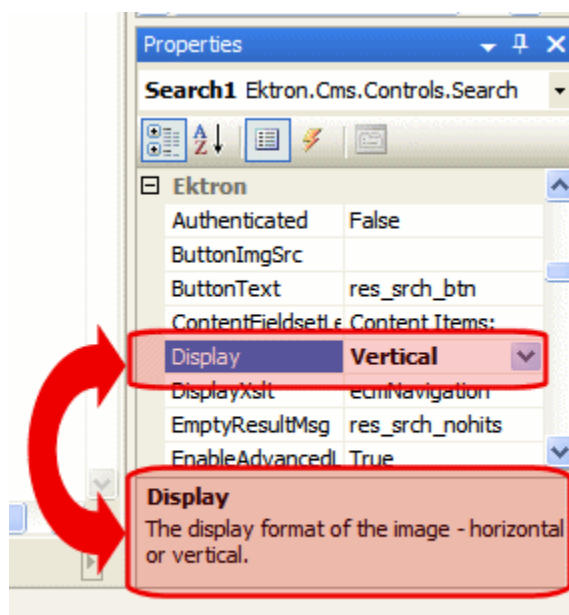
NOTE [Ektron API Documentation](#) is only available inside VS2005 if it was on the server when Ektron CMS400.NET was installed. Otherwise, use the PDF version of the [help](#).

For API Help Inside Visual Studio 2005, follow these steps:

1. In Visual Studio 2005 click **Help > Contents**
2. When the help window appears, expand Ektron CMS400.NET **API Documentation** in the frame on the left.

If you do not see this entry, use the filter and choose **(unfiltered)** or **Ektron Documentation**.

NOTE In addition to the API help, Server Control property descriptions are shown when you highlight a property in design view in VS2005.



API Help in PDF Format

From the Desktop

Click **Start > Programs > Ektron > CMS400 > Documentation > API Reference Manual**

From the Workarea

Click **Settings > Help > Developer's API Reference Manual**

From www.ektron.com

<http://www.ektron.com/web-content-management-solutions.aspx?id=4020>

Ektron Support and Product Pages

- http://www.ektron.com/support/cms400_support.aspx
- <http://www.ektron.com/web-content-management-solutions.aspx>

Ektron's Support and Product pages are a great launch pad to learn about Ektron CMS400.NET.

Supporting Documents

You can find more detailed information about Ektron CMS400.NET in the following manuals:

- Ektron CMS400.NET Administrator Manual
- Ektron CMS400.NET User Manual
- Ektron CMS400.NET Quick Start Manual
- Ektron CMS400.NET Setup Manual

Introduction to Ektron CMS400.NET Server Controls

Ektron CMS400.NET's server controls let you insert many standard methods and properties within the Visual Studio 2005 environment. This means that you can see the effect of your changes in real time -- you don't have to modify a page then compile a sample project to see the results.

You can insert server controls using drag and drop or programmatically. You can also use databinding to retrieve and display data from **Ektron CMS400.NET**. This chapter explains the use of the Ektron CMS400 Server Controls through the following topics.

- "Opening the Sample Project" on page 7
- "Setting Up a Template" on page 9
- "Making the Server Controls Available" on page 10
- "Inserting Server Controls Using Drag and Drop" on page 12
- "Inserting Server Controls Programmatically" on page 16
- "Using Drag & Drop and Programmatically Together" on page 31
- "Data Binding with Server Controls" on page 33
- "Caching with Server Controls" on page 36
- "List of Server Controls" on page 39

Additional information and examples for Ektron Server Controls are available online by going to the Developers section of the sample site that was installed. Here is the path:

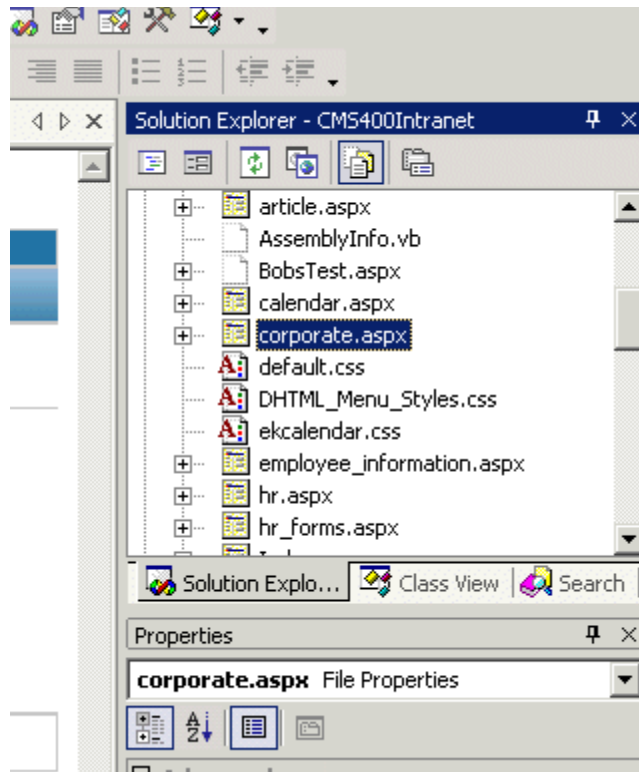
- `http://localhost/cms400demo/Developer/Default.aspx`

Replace `localhost` with the `webroot` where you installed the Example site.

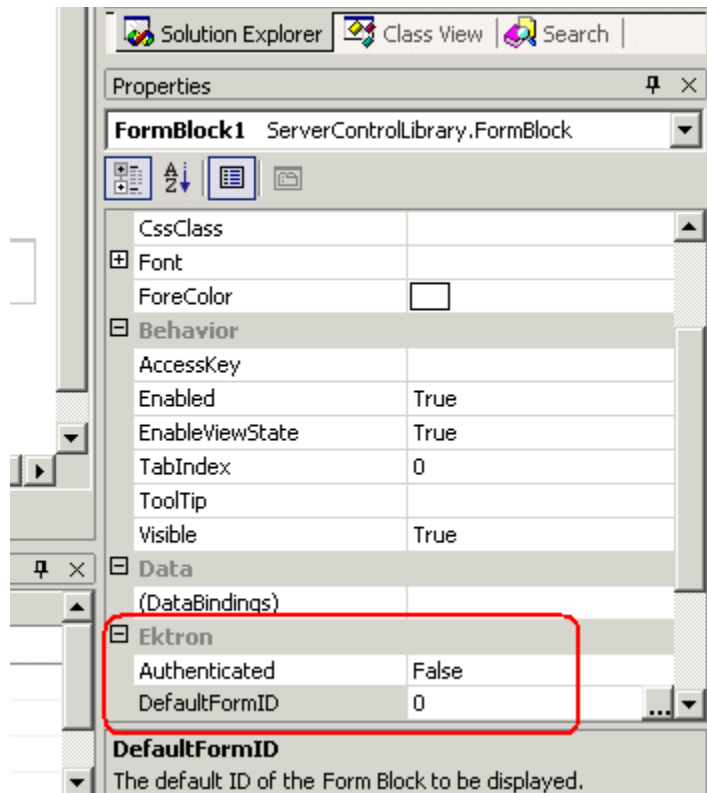
Opening the Sample Project

To help get you started, these directions explain how to open the sample site delivered with **Ektron CMS400.NET**.

1. Browse to and double click Ektron CMS400.NET's solution file, `localhost/cms400demo/cms400demo.sln`.
2. The sample site project opens.
3. To work on a template page, click it from the Solution Explorer.



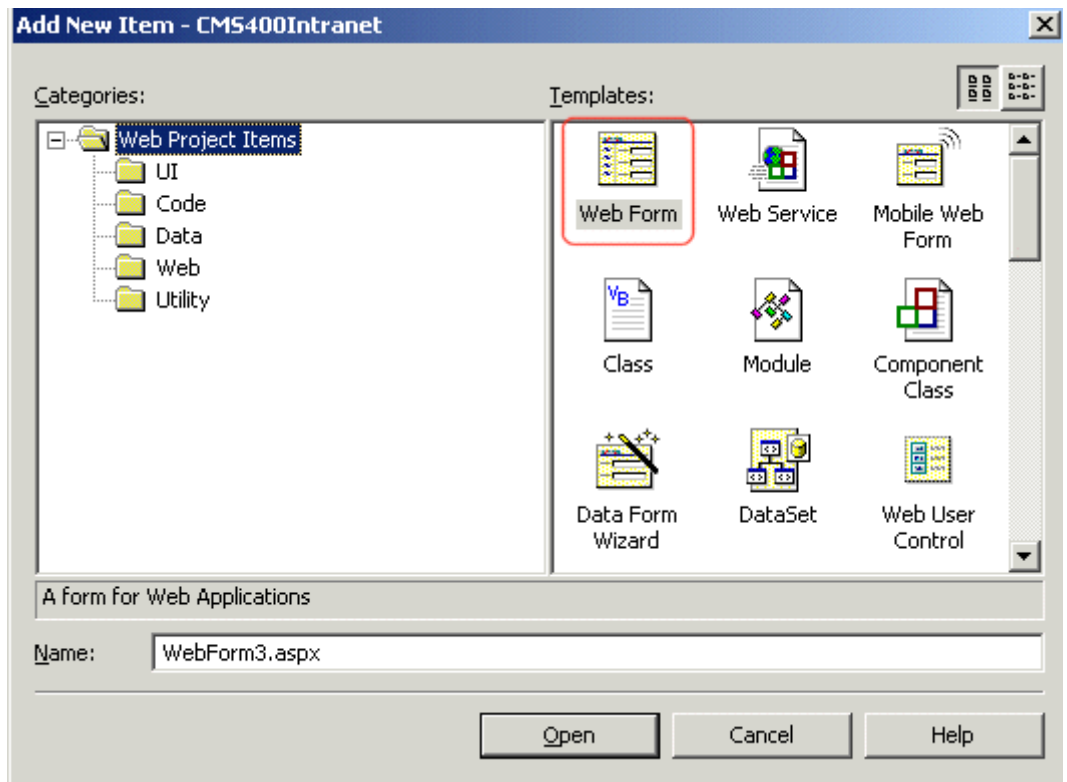
When you select a page, its properties appear in the Properties area (below Solution Explorer), and the page appears in the center of the screen. A control's properties include several standard .NET properties along with Ektron-specific ones. The Ektron properties are labeled as illustrated below.



Setting Up a Template

To create a new template (.aspx) page, follow these steps.

1. Click **Project > Add Web Form**.
2. On the Add New Item screen, click Web Form and assign a name.



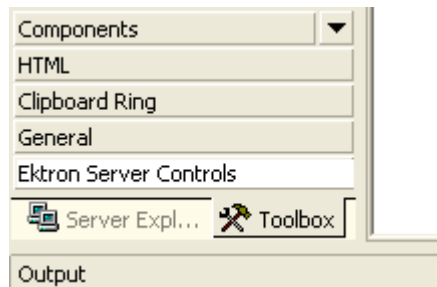
3. Add controls to determine the page content.

Making the Server Controls Available

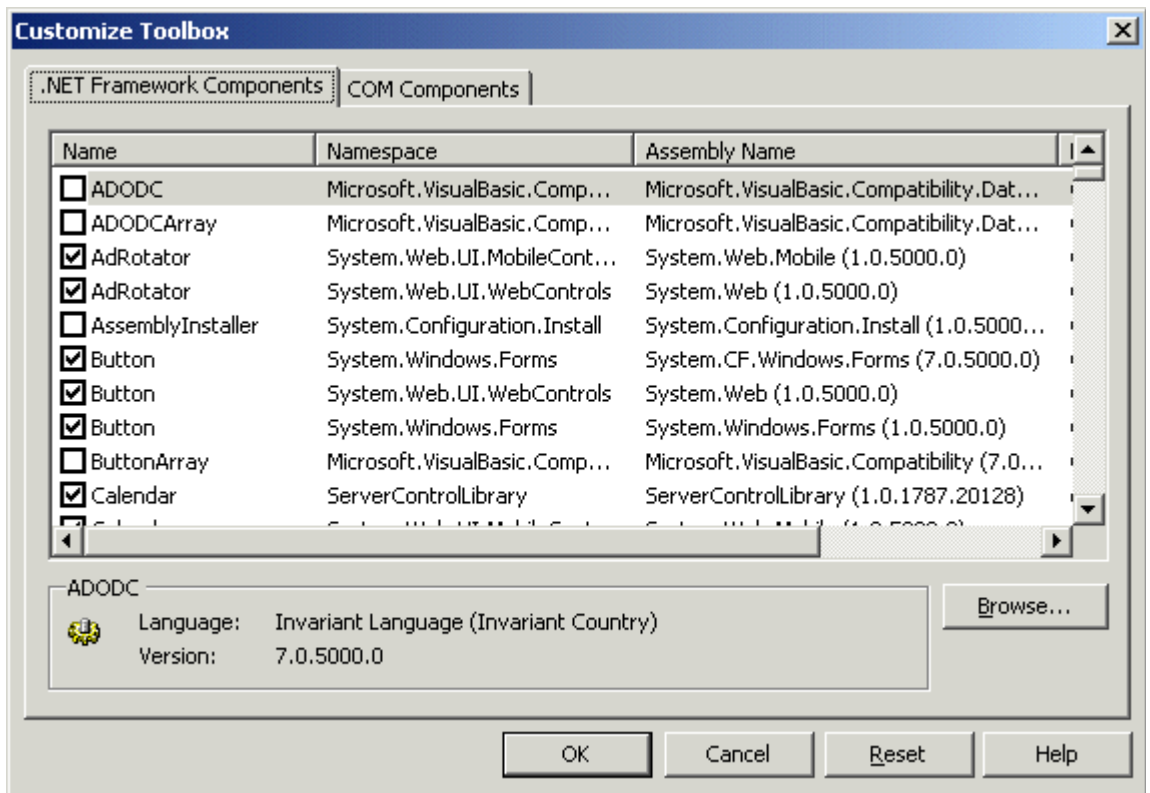
NOTE You must copy the dlls to a local drive before installing them. You cannot copy them from a network drive.

1. Display the Visual Studio 2005 toolbox (**View > Toolbox**).
2. Right click the mouse within the Toolbox.
3. Click **Add Tab**.

4. Type **Ektron Server Controls** then press enter.



5. Click the Ektron Server Controls Tab.
6. Right click the mouse in the empty area.
7. Click **Add/Remove Items**. If using Visual Studio 2005, click **Choose Items**.
8. The Customize Toolbox dialog appears.



9. Select the **.NET Framework Components** tab.

-
10. Browse to the directory that stores Ektron CMS400.NET's dll files, `localhost/cms400demo/bin`, and add the `Ektron.Cms.Controls.dll` file. This file provides access to **Ektron CMS400.NET**'s server controls.
Alternatively, you could use the following location, `C:\Program Files\Ektron\CMS400v60\bin`. The file is identical in both places.
Using the bin folder in your site provides better speed when loading Web pages. However, if you use the bin folder located in Program Files, you do not have to worry about deleting the .dll file if you change or delete your site.
 11. Press **OK**.

Removing the Server Controls

1. Display the Visual Studio 2005 toolbox (**View > Toolbox**).
2. Right click the mouse within the Toolbox.
3. Click **Add/Remove Items**. If using Visual Studio 2005, click **Choose Items**.
4. Click **Namespace** or **Assembly Name** to sort the Server Controls by manufacturer.
5. Uncheck all boxes that are Ektron.Cms.Controls.
6. Click **OK**
7. Right click the Ektron Server Control Tab.
8. Click **Delete Tab** to remove it.

Updating the Server Controls

To update the server controls you must first remove the existing server controls in Visual Basic, then add the new server controls.

For information on removing server controls, see ["Removing the Server Controls" on page 12](#)

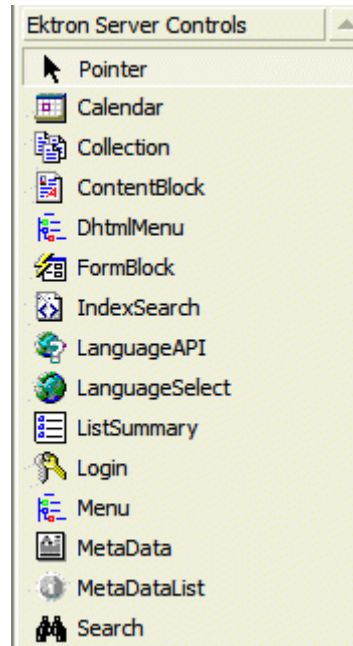
For information on adding server controls, see ["Making the Server Controls Available" on page 10](#)

Inserting Server Controls Using Drag and Drop

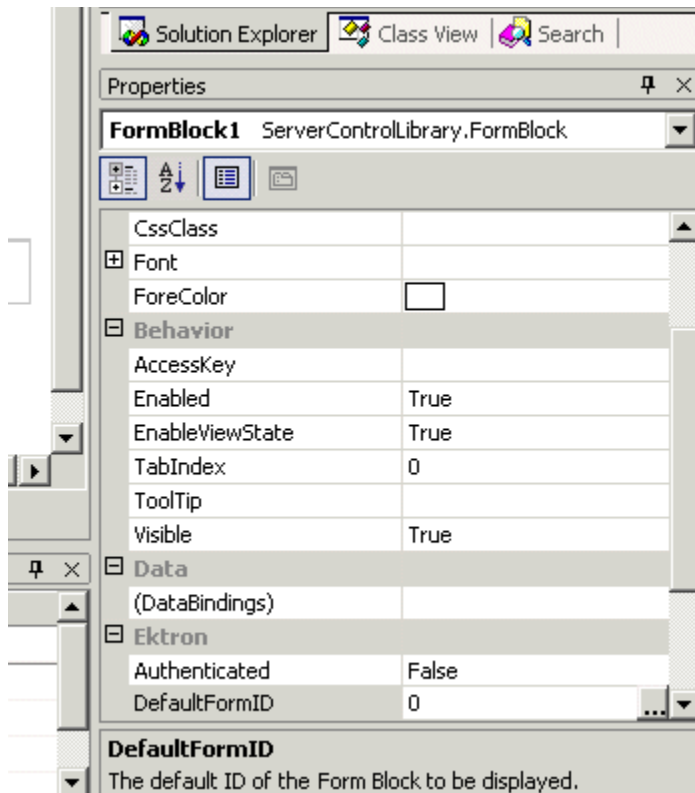
Because Visual Studio 2005 is a visual environment, you can watch the page layout change as you add or move a control and adjust its properties. And, whenever you need to identify an **Ektron CMS400.NET** object (such as content block or collection), you can use the CMS Explorer to visually browse to and select it.

Adding a Server Control to a Template

1. Display the Visual Studio 2005 toolbox (**View > Toolbox**).
2. Click the **Ektron Server Control** tab. **Ektron CMS400.NET**'s server controls appear.



3. Drag a server control and drop it into the desired location on the page.
4. As desired, modify the control's properties using the Properties area of Visual Studio 2005.



For a description of server control properties, see ["List of Server Controls" on page 39](#).

ServerControlWS.asmx

Best Practice

For improved security, Ektron recommends renaming or removing the Web services file when you move it to your production server. After installation, this file is named **ServerControlWS.asmx** and resides in the `/cms400demo/Workarea/` folder in your Web root.

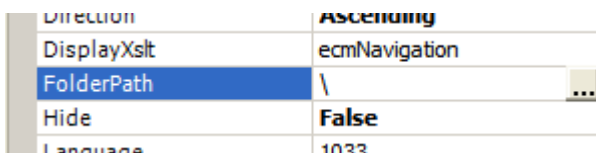
ServerControlWS.asmx is the Web service that lets the server controls talk to **Ektron CMS400.NET**. The path is coded in the web.config file. This is how it appears:

```
<!-- Web Service URL for Server Controls design time --> <add key="WSPath" value="http://localhost/CMS400Demo/Workarea/ServerControlWS.asmx" />
```

You will need to edit this line if you change the location or name of the **ServerControlWS.asmx** file.

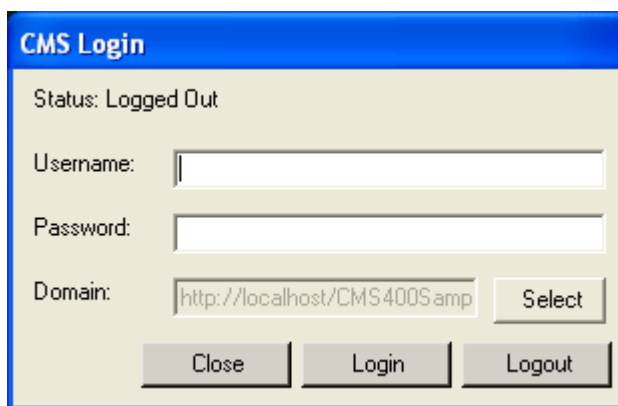
Using CMS Explorer to Browse Your Ektron CMS400.NET Site

Ektron CMS400.NET provides a window (called the CMS Explorer) that lets you browse through your Web site to identify CMS objects such as folders, calendars, blogs and content blocks. For example, if you insert a List Summary Server Control, its **FolderID** property identifies the folder whose contents are displayed. If you are not sure of the folder path, click the gray button.

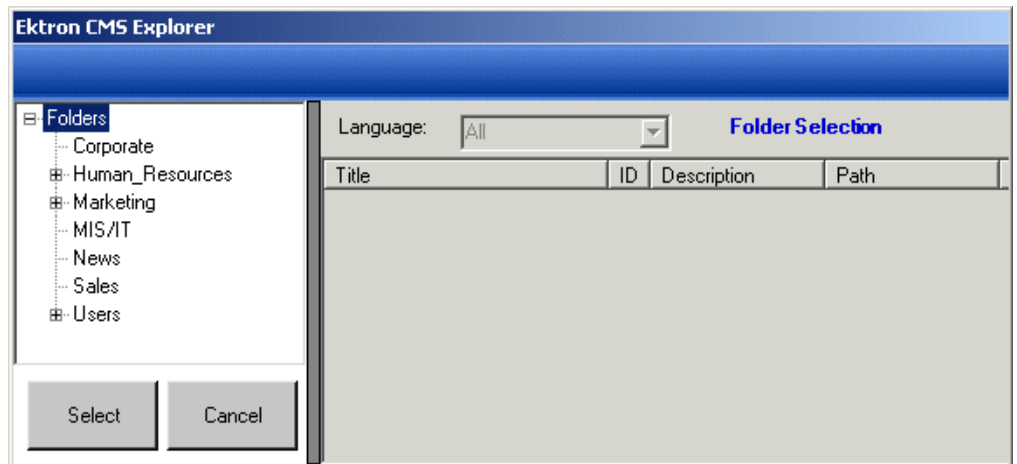


When you do, a login screen appears (illustrated below).

NOTE A user cannot be logged into the CMS Explorer and the CMS400.NET Workarea at the same time. If you log into the CMS Explorer while logged into the Workarea, you will receive the following error when you try to use the Workarea: Error: The current user does not have permission to carry out this request [CBRR31]. If you log into the Workarea while logged into the CMS Explorer, you will need to re-login to the CMS Explorer when you return to using it.

A screenshot of a 'CMS Login' dialog box. The title bar is blue and says 'CMS Login'. The main area has a light beige background. At the top, it says 'Status: Logged Out'. Below that are three input fields: 'Username:', 'Password:', and 'Domain:'. The 'Domain:' field contains the text 'http://localhost/CMS400s.asp' and has a 'Select' button to its right. At the bottom, there are three buttons: 'Close', 'Login', and 'Logout'.

Enter your **Ektron CMS400.NET** username and password. When you do, the CMS Explorer window appears, providing a visual way to browse your **Ektron CMS400.NET** Web site.



Navigate through the folders, select an object, then click the **Select** button. The selected object is pasted to the server control property.

NOTE Although you see the object in the selected language in Visual Studio 2005, the language is not stored. For example, if you select a German content block whose ID=2, **Ektron CMS400.NET** only stores content block ID=2. When a visitor to your site browses to that page, the content block does not appear in the selected language. Instead, it is determined by a cookie or the user's language selection.

Build the Project

After you open the solution file in Visual Studio 2005 and add the required files, build the project. This may take a while the first time you do so. After the project is built, a browser opens and it is displayed as a Web page.

You can also view a Web page while working on it by right clicking on the Web form and clicking **View in Browser**.

Inserting Server Controls Programmatically

Besides using the drag-and-drop method to insert a server control (explained in ["Inserting Server Controls Using Drag and Drop" on page 12](#)), you can insert one programmatically. You might prefer to insert server controls programmatically for several reasons, such as

- you want the control to be loaded into memory only under certain conditions. In this case, insert the logic that only displays the control if the condition exists.
- you want to display only certain properties of an object, such as the title of last edited date of a content block.

Note that you can drag and drop controls on your page then customize them using code behind. For more information, see: ["Using Drag & Drop and Programmatically Together" on page 31](#).

The following sections explain how to use the server controls programmatically.

- ["Steps to Insert a Server Control Programmatically" on page 17](#)
- ["Server Control Properties in Code Behind" on page 20](#)
- ["Customizing Server Controls" on page 29](#)

Steps to Insert a Server Control Programmatically

To insert an Ektron server control programmatically, follow these steps.

["Step 1: Declare the Namespace" on page 17](#)

["Step 2: Create an Instance of the New Control" on page 18](#)

["Step 3: Set the Properties" on page 18](#)

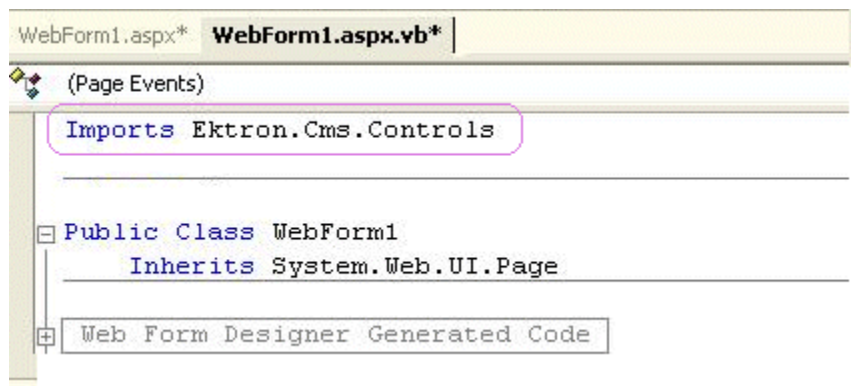
["Step 4: Call the Fill Method" on page 19](#)

["Step 5: Use .text Property to Display the Control" on page 19](#)

Each step is described below. These steps show an example of using the Collection Server Control.

Step 1: Declare the Namespace

Before inserting the server control, you can declare the namespace at the top of the code behind Visual Basic file, as shown below.



NOTE You do not need to declare a namespace. However if you do not, you must fully qualify objects that you create.

Best Practice

For any customization of the CMS, classes or controls that inherit from Ektron classes, it is recommend that you create your own namespace within 'Ektron.Cms.Custom'. For example, if your company is 'AcmeExampleTech, Inc.' you should create all of your custom classes within the namespace 'Ektron.Cms.Custom.AcmeExampleTech'.

Step 2: Create an Instance of the New Control

Next, declare a control as an object in the code behind. In this example, we create a collection named `MyColl`.

```
dim MyColl as New Collection
```

or

```
dim MyColl as new Ektron.Cms.Controls.Collection
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Collection MyColl = new  
Ektron.Cms.Controls.Collection();
```

You can declare any server control as an object by using the server control name in the ["List of Server Controls" on page 39](#). Another example would be: `Dim MyMdl as New MetaDataList`.

Step 3: Set the Properties

NOTE For descriptions of the properties and how to use them, see ["Server Control Properties in Code Behind" on page 20](#).

After inserting the control, you can set the properties that you want to display on the page. For example:

```
Dim MyColl as New Collection  
MyColl.DefaultCollectionID = 4
```

```
MyColl.Page = Page
```

or if you are using code behind to insert the control,

```
dim MyColl as new Ektron.Cms.Controls.Collection  
MyColl.DefaultCollectionID = 4
```

```
MyColl.Page = Page
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Collection MyColl = new  
Ektron.Cms.Controls.Collection();  
MyColl.DefaultCollectionID = 4;
```

```
MyColl.Page = Page;
```

These lines tell the page to display CollectionID 1 unless otherwise specified.

IMPORTANT!

When using code behind to add a server control to your Web form, you must set the Page object for the server control to Page. For example, `MyColl.Page = Page`. This line needs to appear between Dim new server control line and the `Fill()` line. This line is not added when dragging and dropping a server control on a Web form. See Also: "Referencing the Page Property" on page 28.

NOTE

If you do not know an object's ID number, you can switch to Design mode, drag and drop the object, then use the CMS Explorer to find the ID number. (See "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.) If you do this, remember to delete the dropped object when you are done.

You can also obtain the ID number via the Workarea.

```
Dim MyColl as New Collection
MyColl.DefaultCollectionID = 4
MyColl.Page = Page
MyColl.Random = True
```

This line sets the Random property to **true**. See Also: "Random" on page 139.

Step 4: Call the Fill Method

Because there is no render event when using objects as components not as controls, use the Fill method to fill an object's properties on the page. An example is below.

```
Dim MyColl as New Collection
MyColl.DefaultCollectionID = 4
MyColl.Page = Page
MyColl.Random = True
MyColl.Fill()
```

Step 5: Use .text Property to Display the Control

Finally, use the `.text` property to determine what appears on the Web page. For example, to display the first item in a collection, use this syntax.

NOTE

Before adding this line you need to drag and drop a label on your Web form.

```
Dim MyColl as New Collection
MyColl.DefaultCollectionID = 4
MyColl.Page = Page
MyColl.Random = True
MyColl.Fill()
Label1.Text = myColl.EkItems(0).Title
```

To display *all* items in a collection, use this syntax.

```
Dim myColl As New Ektron.Cms.Controls.Collection
Dim ekitem As New Ektron.Cms.Common.ContentBase
myColl.DefaultCollectionID = 2
myColl.Page = Page
myColl.Fill()
Label1.Text = "<ul>"
For Each ekitem In myColl.EkItems
```

```
        Label1.Text &= "<li><a href="" & ekitem.QuickLink &  
"">" & ekitem.Title & "</a>"  
        Next  
        Label1.Text &= "</ul>"
```

This example displays the quick link for every content block in the collection, formatted as a bulleted list. You can use similar code to display a List Summary or search results. The following explains the new (red) code above.

- `label1.Text = ""` displays the opening tag for the bulleted list
- `For Each ekitem In myColl.EkItems` creates a loop for all content blocks in the collection
- `label1.Text &= "" & ekitem.Title & ""` for each content block in the collection, displays its quicklink and title
- `Next` loops through all content blocks in the collection
- `label1.Text &= ""` closes the bulleted list

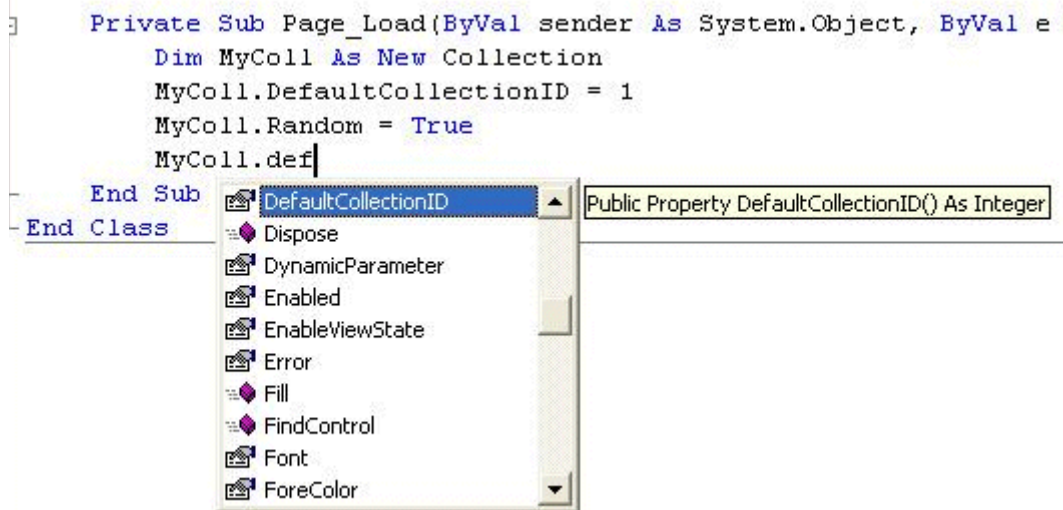
Server Control Properties in Code Behind

Every server control has properties associated with it that you can only access programmatically. This subsection explains, what they are, how to access them, and how to use them.

- ["Displaying Properties for a Control" on page 20](#)
- ["Mutual Server Control Properties in Code Behind" on page 21](#)
- ["Accessing Additional Properties" on page 23](#)
- ["Accessing Items in an Array" on page 28](#)
- ["Referencing the Page Property" on page 28](#)

Displaying Properties for a Control

You can use Visual Studio 2005's intellisense feature to display a control's properties. The intellisense box appears as soon as you insert the period (.) after the object, as illustrated below.



The intellisense box displays *all* properties that can be applied. For a description of Ektron's standard server control properties, see ["List of Server Controls" on page 39](#). To learn about native VS 2005 properties, see its documentation.

The property's tooltip text indicates its type. In the above example, you can see that the DefaultCollectionID's type is integer.

For more information about accessing **Ektron CMS400.NET** object properties, see ["Customizing the Server Control in the Code Behind" on page 30](#).

Mutual Server Control Properties in Code Behind

There are several read-only properties that can be called in the code behind and used by every server control. The list below explains these properties.

NOTE The properties do not display values within Visual Studio.NET during design time. Instead, they only display values at run time, which are dependent on the user's login status.

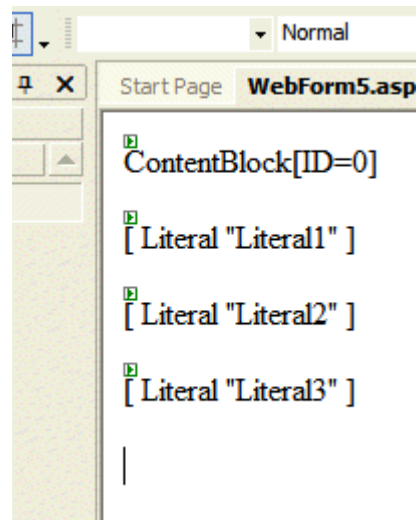
Property	Value	Data Type
IsLoggedIn	Tells if a user is logged in to Ektron CMS400.NET . True = User is logged in False = User is not logged in	Boolean
LoggedInUserName	Gets the Ektron CMS400.NET user name to display.	String

Property	Value	Data Type
LoggedInUserID	Gets the Ektron CMS400.NET ID of the user to display.	Integer

These properties will allow you to personalize any page with your users names and IDs, and show if they are logged in. Here is an example of using these properties in code behind.

NOTE You must be logged in to **Ektron CMS400.NET** for this example to show your name and ID.

1. Drag an **Ektron CMS400.NET** server control onto a Web form.
2. Drag three Literals onto the Web form.

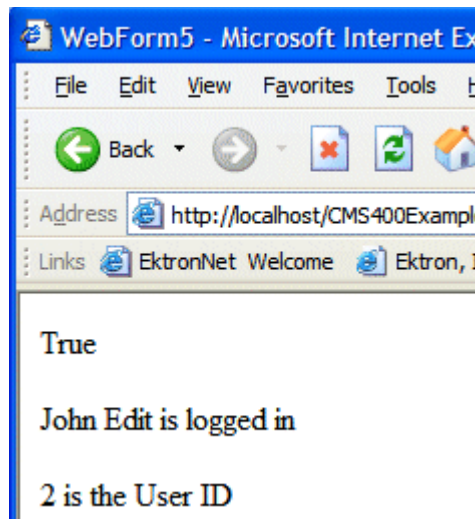


3. Open the code behind for the Web form.
4. Add the following code to the Page_Load event.

```
Literal1.Text = ContentBlock1.IsLoggedIn
If ContentBlock1.IsLoggedIn Then
    Literal2.Text = ContentBlock1.loggedInUserName & " is logged in "
    Literal3.Text = ContentBlock1.loggedInUserID & " is the User ID "
End If
```

5. Build and run the solution.
6. Browse to the login page and log in.
7. Browse to the new Web form you added.

8. The login information is displayed.



Accessing Additional Properties

In addition to the standard properties, **Ektron CMS400.NET** provides access to additional properties for the following objects.

- ListSummary
- Collection
- Search
- ContentBlock
- FormBlock

To access additional properties, use the same syntax you use for standard properties but add `.ekitem` or `.ekitems` after the object. Here is an example.

```
dim MyCB as New ContentBlock
MyCB.DefaultContentID = 30
MyCB.Page = Page
MyCB.Fill()
label1.text = MyCB.EkItem.dateCreated
```

or

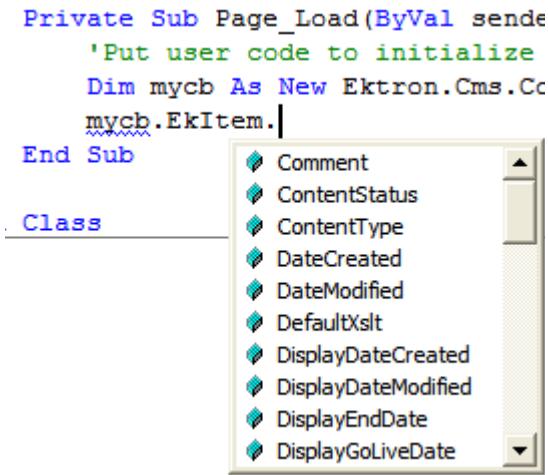
```
dim MyCB as new Ektron.Cms.Controls.ContentBlock
MyCB.DefaultContentID = 30
MyCB.Page = Page
MyCB.Fill()
label1.text = MyCB.EkItems.dateCreated
```

IMPORTANT!

To access additional properties for the Collection, ListSummary, and Search objects, use `ekitems`, not `ekitem`. For example:

`MyColl.ekitems(0).dateCreated`, where (0) is the index of the array. For more Information on using ekitems, see "Accessing Items in an Array" on page 28.

You can use intellisense to select from a list of additional object properties, as shown below.



The additional properties are listed below.

NOTE The following properties are read-only. For example, you can get a content block's ID and pass it through to another part of the code, however you cannot set a content blocks ID to be shown. `mycb.Ekitem.id = 8` will not set a content block's ID. The correct way to set a content blocks ID is `DefaultContentID = 8`.

Property	Description	For more information, see
Comment	The content block's comment	
ContentStatus	The status of the content block. <ul style="list-style-type: none">approvedchecked outchecked inexpiredpending deletionpending expirationpending start datesubmitted	Ektron CMS400.NET User Manual, "Appendix A: Content Statuses."

Property	Description	For more information, see
ContentType	One of the following <ul style="list-style-type: none"> all types archived content archived forms content forms 	To learn about archived content, see the Ektron CMS400.NET User Manual, “Scheduling Content to Begin and End” > “Setting Archive Options.”
DateCreated	The date when the content block was created, formatted as a .NET date type	
DateModified	The date when the content block was modified, formatted as a .NET date type	
DefaultXslt	The default Xslt	
DisplayDateCreated	The date when the content block was created. It is formatted as a string that represents Ektron CMS400's display of the date.	
DisplayDateModified	The date when the content block was edited. It is formatted as a string that represents Ektron CMS400's display of the date.	
DisplayEndDate	The content block's end date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, “Scheduling Content to Begin and End” > “Setting an End Date on a Content Block”
DisplayGoLiveDate	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, “Scheduling Content to Begin and End” > “Setting a Start Date”
DisplayStartDate	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, “Scheduling Content to Begin and End” > “Setting a Start Date”

Property	Description	For more information, see
EndDate	The content block's end date, formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting an End Date on a Content Block"
EndDateAction	<ul style="list-style-type: none"> archive display archive expire refresh report 	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting Archive Options."
FolderID	The ID of the folder that contains each content block	
GoLiveDate	The content block's start date formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"
Html	The content that makes up the content block. If content block is in XML it will return it as raw XML content	
Hyperlink	Content block title wrapped by <a href> tags	
Id	The content block 's ID number	
InheritedFrom	If folder permissions are inherited, the folder from which they are inherited	Ektron CMS400.NET Administrator manual "Setting Permissions" > "Inheritance"
IsInherited	Whether a content block's permissions are inherited	Ektron CMS400.NET Administrator manual "Setting Permissions" > "Inheritance"
IsPrivate	Whether or not a content block is private	Ektron CMS400.NET Administrator manual "Managing Users & User Groups" > "Private Content & Memberships"
Language	The content block's language	Ektron CMS400.NET Administrator manual "Multi-Language Support" > "Working with Multi-Language Content"

Property	Description	For more information, see
LastEditorFname	The first name of the last person to edit the content block	
LastEditorLname	The last name of the last person to edit the content block	
PackageDisplayXSLT	If the content block is XML, the name of its XSLT	
QuickLink	The content block's quicklink	Ektron CMS400.NET User manual "Library Folder" > "Quicklinks"
StartDate	The content block's start date formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"
Status	The status of the content block <ul style="list-style-type: none"> • approved • checked out • checked in • expired • pending deletion • pending expiration • pending start date • submitted 	Ektron CMS400.NET User manual "Appendix A: Content Statuses"
Teaser	The content block summary	"Adding a Content Block Summary" in the Ektron CMS400.NET User manual
TemplateLink	Currently empty and not being used with the ContentBlock server control	
Title	The content block title	
UserID	Last user who edited the content	

Property	Description	For more information, see
Xslt1	The content block's first Xslt, as defined in Ektron CMS400.NET	
Xslt2	The content block's second Xslt, as defined in Ektron CMS400.NET	
Xslt3	The content block's third Xslt, as defined in Ektron CMS400.NET	
Xslt4	The developer can use this property programmatically. Ektron CMS400.NET only uses Xslt1, 2 and 3 in the workarea.	
Xslt5	The developer can use this property programmatically. Ektron CMS400.NET only uses Xslt1, 2 and 3 in the workarea.	

Accessing Items in an Array

To access and manipulate content blocks returned by an object, use the common class `Ektron.Cms.Common.ContentBase`. *EkItems* is an array of `Ektron.Cms.Common.ContentBase`. *EkItem* is a single `Ektron.Cms.Common.ContentBase`.

Search, Collection and ListSummary have *EkItems* (an array of `ContentBase`), while *ContentBlock* has a single *EkItem*. Here is an example of how to use this feature.

```

dim MyC as new Ektron.Cms.Controls.Collection
MyC.DefaultCollectionID = 1
MyC.Page = Page
MyC.Fill()
dim item as Ektron.Cms.Common.ContentBase
MyC.Text = "<ul>"
for each item in MyC.EkItems
    MyC.Text &= "<li>" & item.Title & "</li>"
next
MyC.Text &= "</ul>"
Response.Write(MyC.Text())

```

This example formats every item in the collection in a bulleted list.

Referencing the Page Property

Server controls require a reference to their parent page (for example, utilizing the `DynamicParameter` property on a content block to check for a query string), you must provide access to the page object if you declared your control in the code

behind. To do this, set the control's page property to the Web page you're working on. For an example, see the code in red and italics below.

```
dim search as new Ektron.Cms.Controls.Search()  
MySearch.Page = Page  
MySearch.Fill()
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Search MySearch = new  
Ektron.Cms.Controls.Search();  
MySearch.Page = Page;  
MySearch.Fill();
```

This relationship is only required when inserting a control in the code behind. When dragging and dropping, even if you make changes in code behind, the relationship is automatically generated.

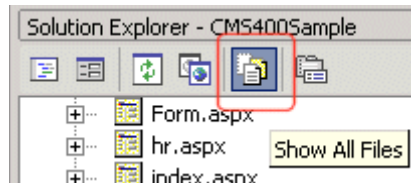
Best Practice

Ektron recommends including the page property reference when using the server controls as components in the code behind.

Customizing Server Controls

One of Visual Studio 2005's strengths is its separation of coding and logic from presentation. Web page formatting is handled by a page's HTML, while the logic is handled by the code behind, which is stored in the corresponding .vb file. For example, if the ASP.NET page is mypage.aspx, the code behind file is mypage.aspx.vb.

NOTE If you do not see the code behind files, click the Show All Files button on the VS 2005 Solution Explorer toolbar.



Within the vb file, you can use Visual Basic to insert code to manipulate the events that occur on the page.

This subsection contains the following topics:

- "Recognizing the Server Control Within the HTML" on page 30
- "Recognizing the Server Control Within the Code Behind" on page 30
- "Customizing the Server Control in the Code Behind" on page 30
- "Troubleshooting Error Creating Control Message" on page 31

Recognizing the Server Control Within the HTML

Within a Web page's HTML, a <cms> tag wraps the VS 2005 object, as shown below.

```
<cms:Search id="Search1" runat="server" ButtonText="Search" Display="Vertical"></cms:Search>
```

Above is an example of a Search Server Control. Below is an example of a ContentBlock Server Control.

```
<cms:ContentBlock id="ctrlMainContentBlock" runat="server" DefaultContentID="1"
DynamicParameter="id" OverrideXslt="Default"></cms:ContentBlock>
```

Recognizing the Server Control Within the Code Behind

Within the VS 2005 code behind file, the Ektron server controls appear (along with the VS 2005 controls) in the Web Form Designer Generated Code section. When you click the plus sign (+) to display this section, you see something like the following. The content block listed in HTML above is circled below to help you see their relationship.

```
#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()...
    Protected WithEvents MetaDataArea As System.Web.UI.WebControls.Literal
    Protected WithEvents DhtmlJavaScript As System.Web.UI.WebControls.Literal
    Protected WithEvents DropDownMenu As System.Web.UI.WebControls.Literal
    Protected WithEvents ctrlMainContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlTopContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlBottomLeftContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlBottomRightContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlLanguageSelect As Ektron.Cms.Controls.LanguageSelect
    'NOTE: The following placeholder declaration is required by the Web Form Designer
    Private designerPlaceholderDeclaration As System.Object
```

The next section of the code behind page loads the page into the browser.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
```

We want our events to occur while the page is loading, so we'll add custom code following this line.

Customizing the Server Control in the Code Behind

To customize an Ektron server control in the code behind, insert code similar to the following *after* the Page_load command.

```
Dim MyObj As New Ektron.Cms.Controls.ContentBlock
```

This code declares a variable named MyObj and assigns to it the value of a content block. The content block is part of the Ektron.CMS.Controls content base, so it has access to the **Ektron CMS400.NET** database.

After defining MyObj as a content block, you can access its properties. For example, to assign a defaultID of 24, insert the following.

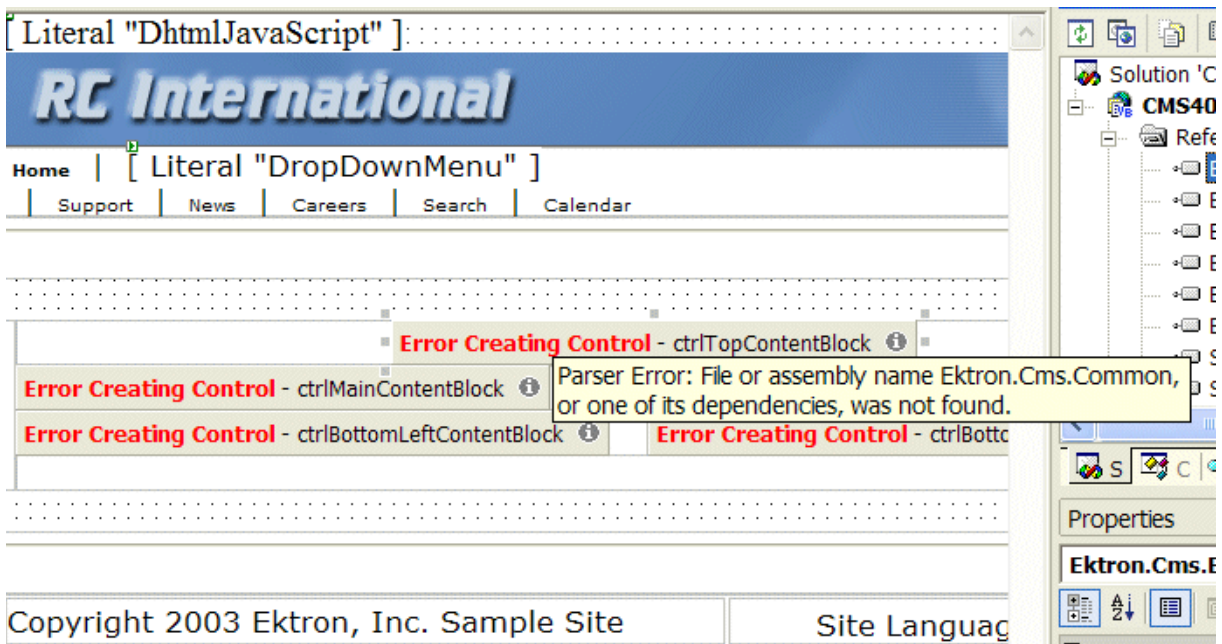
```
Dim MyObj As New Ektron.Cms.Controls.ContentBlock
MyObj.DefaultContentID = 24
```

Now, the content block can be specified dynamically in the URL of the hyperlink that calls it. If not, content block 24 displays.

This is just an example of programmatically applying property values to content blocks. For a complete list of properties available to **Ektron CMS400.NET** objects, see ["List of Server Controls" on page 39](#).

Troubleshooting Error Creating Control Message

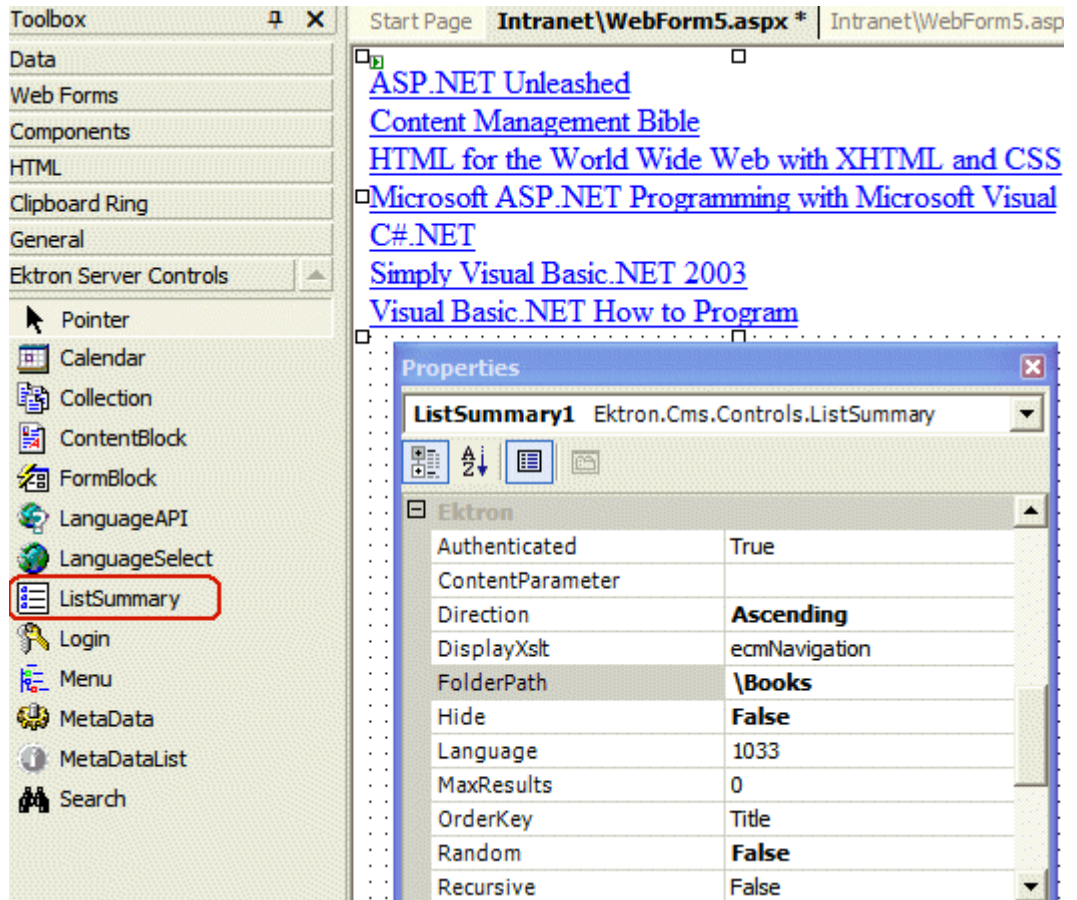
If you get an Error Creating Control message while trying to use a server control, you can view the text of the error message by hovering the mouse over the control. See illustration below.



Using Drag & Drop and Programmatically Together

All server controls can be further manipulated after dragging and dropping them on a web form by using the code behind. Here is an example of using a drag and drop ListSummary Server Control then modifying it programmatically.

First, drag and drop a List Summary on your Web form and set your properties.



Then add the following code to the code behind.

```
Dim myString As String
Dim i
For i = LBound(ListSummary1.EkItems) To UBound(ListSummary1.EkItems)
myString &= "<a href=""" & ListSummary1.EkItems(i).QuickLink & """" >" &
ListSummary1.EkItems(i).DateCreated & "</a><br>"
ListSummary1.Text = myString
Next
```

An Explanation of the code:

ListSummary1 is the ID of the object. It is used to get access to its properties.

Create a string that contains the output (myString).

```
Dim myString As String
```

Set the object's text property to that string.

```
myString &= "<a href=""" & ListSummary1.EkItems(i).QuickLink & """" >" &
& ListSummary1.EkItems(i).DateCreated & "</a><br>"
```

Wrap in a loop so it loops through each of the items.

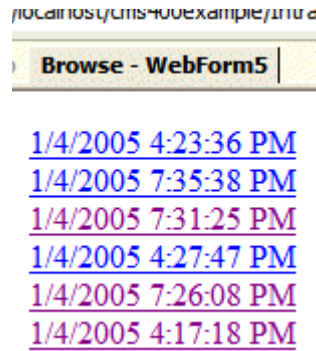
```

Dim i
For i = LBound(ListSummary1.EkItems) To UBound(ListSummary1.EkItems)

Next

```

This example outputs the date created for each content block in a List Summary.



Data Binding with Server Controls

With data binding, you can bind Ektron server controls to a GridView Control, DataList Control, or Repeater Control. This gives more flexibility when you use data from the Ektron server controls. Some benefits of data binding are ease of data manipulation and the ability to format data.

Title	ID Teaser
Plastic Molder #123	13 RC International is looking for an experienced plastics molder.
Servo Control Engineer #124	14 RC International is looked for a highly skilled servo control eng

The following are data bindable Ektron Server Controls.

- Collection
- Content Block
- Form Block
- List Summary
- Search

The following is an example of code behind that uses a GridView to display a Collection:

```

Dim myCol As New Ektron.Cms.Controls.Collection
myCol.DefaultCollectionID = 4
myCol.Page = Page
myCol.Fill()

```

```
GridView1.DataSourceID = myCol  
GridView1.DataBind()
```

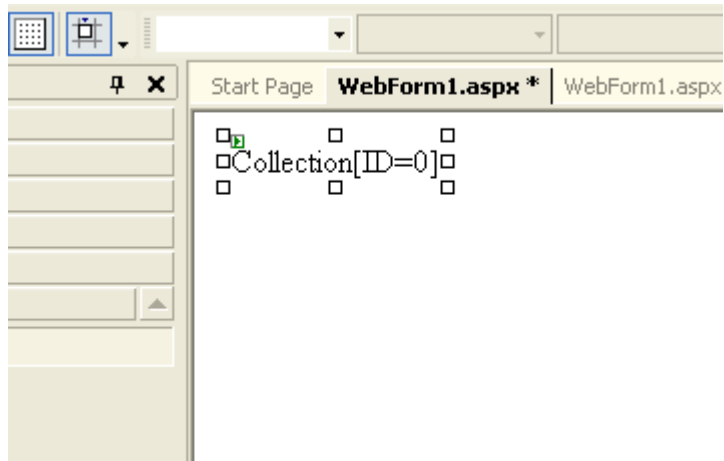
This subsection contains the following topics:

- ["Steps to Data Binding Using Drag and Drop Server Controls" on page 34](#)

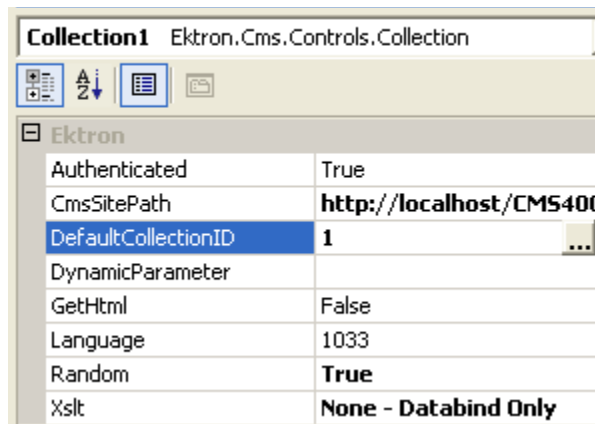
Steps to Data Binding Using Drag and Drop Server Controls

The following example uses a Collection.

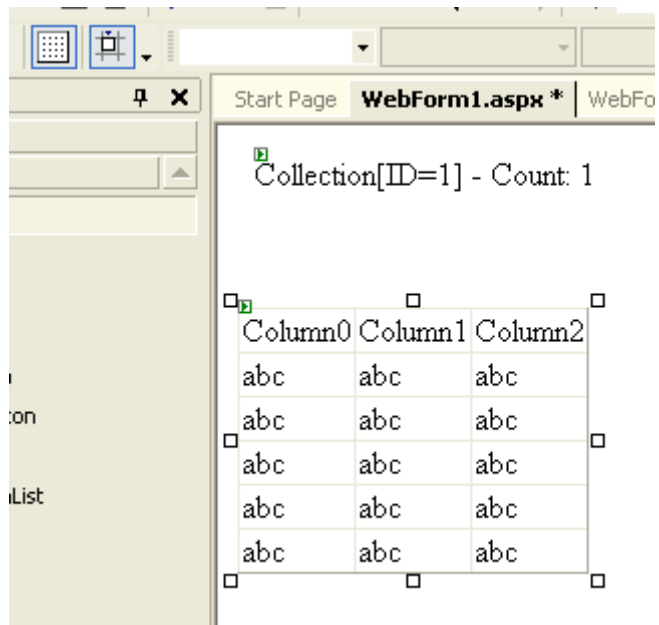
1. Drag and drop a data bindable Server Control on the Web form.



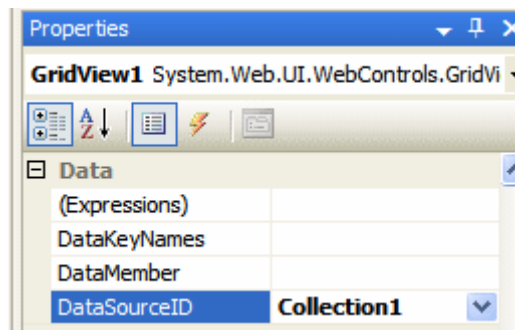
2. In properties, choose the Ektron default ID for the Server Control and make sure the XSLT is set to **None- DataBind Only**.



3. Drag and drop a GridView on the Web form.



4. In the GridView properties, choose the DataSourceID.



5. The GridView is populated with column headers and the type of information that will be in the table.

-
6. Right-click on the GridView. Then, choose **View in Browser**. The table appears.

LastEditorFname	DateModified	Html	QuickLink	HasErrors	Teaser	ID
Application	2/17/2004 4:12:11 PM		/CMS400Sample/products.aspx?id=5	False	RC Internationals most successful entry in to the competitive RC racing world is the RC Cheetah.	5

For more information on GridView, DataList, Repeater and DetailsView see the help inside Visual Studio.

Caching with Server Controls

One of the most important factors in building high-performance, scalable Web applications is the ability to store items in memory the first time they are requested. These items include data objects, pages, or parts of a page.

Known as caching, this process allows you to use a number of techniques to store page output or application data across HTTP requests and reuse it. You can store items on the Web server or other software in the request stream, such as the proxy server or browser.

With caching, the server does not have to recreate information, particularly information that demands significant processor time or other resources. This saves valuable time and resources.

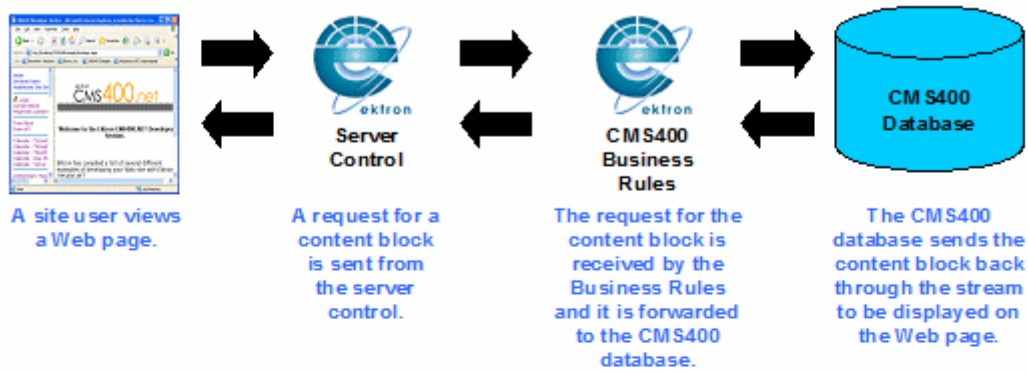
There are two types of caching available:

- [Caching Individual Content](#)
- [Page Level Caching](#)

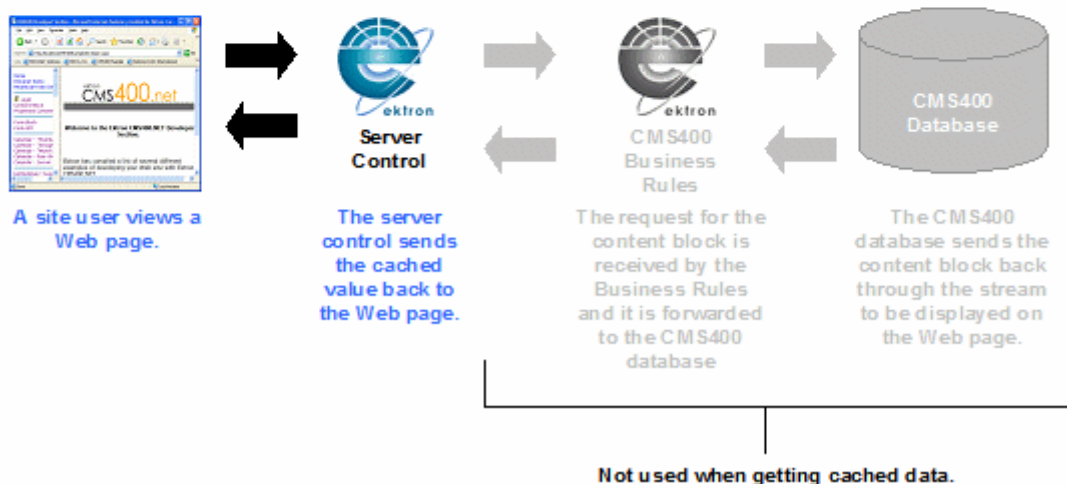
Caching individual content allows you to cache a part of a Web page and is only available on some server controls. With Page Level Caching, you can use any server control and the entire page is cached.

The following is a visual example of how data flows in non-cached and cached environments.

Viewing a Site without Caching



Viewing a Site with Caching (After an initial view)



IMPORTANT!

Controls serve data from the cache only if the user is not logged in.

Caching Individual Content

Caching individual content is available on the following server controls:

- ContentBlock
- FormBlock
- Collections

-
- ListSummary
 - MetadataList
 - Menu
 - DhhtmlMenu

Before caching individual content, you must make sure the `ek_CacheControls value = "1"` (one). This is the default. This setting is found in the `Web.config` file and allows you to control whether each server control can utilize caching.

Once caching is turned on, use the `CacheInterval` property in the server control to set the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the `CacheInterval` property to 300 (three hundred).

WARNING!

If you change `ek_CacheControls` value to 0 (zero), you will not be able cache individual content. This also causes the `CacheInterval` property of each server control to be set to 0 (zero) the next time that server control is called on your site. This is done so you can globally reset caching of Web site pages to 0 (zero). For example, if your site has five hundred Web pages and half of them use caching, you can set the `ek_CacheControls` to 0 (zero) and then none of them will have caching.

To set up caching of individual content for any Web page, follow these steps.

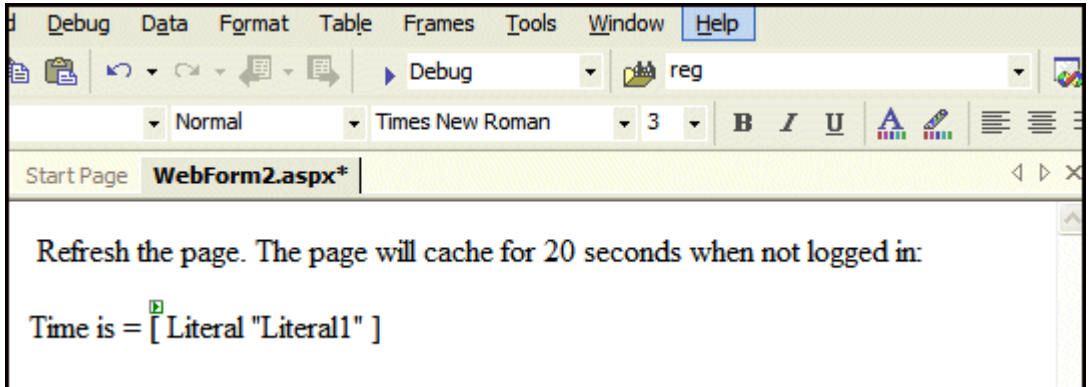
1. Make sure the `ek_CacheControls value = "1"` in the `Web.config` file.
2. Add a server control that supports caching of individual content to a Web form.
3. Set the `CacheInterval` property in the server control to the length of time for the data to be cached.

Page Level Caching

Below is an example of a page level cache for non logged in users. In this example, you use a Server Control to define whether a user is logged in or not. Then, further define if the time is cached, based on the user's status.

1. Create a new Web form in your Ektron CMS400.NET project.
2. Add the following text to the Web form:
Refresh the page. The page will cache for 20 seconds when not logged in:
Time is =

3. Next to **Time is =**, add a literal.



4. Below that, add a ContentBlock Server Control.
5. Set the DefaultContentID. For this example, DefaultContentID = 20.
6. Add the following code to the code behind.

```
If Not ContentBlock1.IsLoggedIn Then
    Response.Cache.SetExpires(DateTime.Now.AddSeconds(20))
    Response.Cache.SetCacheability(HttpCacheability.Public)
    Response.Cache.SetValidUntilExpires(True)
    Response.Cache.VaryByParams("id") = True
    Response.Cache.SetVaryByCustom("cmsCache")
End If
Literal1.Text = Now()
```

7. Build your Web form.
8. Browse to your Web form with your browser.
9. Press the refresh button on your browser.

If you are not logged in to CMS400.NET, the time remains the same for twenty seconds. After twenty seconds, when you refresh, you get the new time.

NOTE The same code can be employed in a user control to cache the output in a particular region of the page.

List of Server Controls

To customize your Ektron CMS400.NET application, you can use these server controls.

Server Control Names	Description	More Information
AnalyticsTracker	Track statistics about visits to your Web site.	"Content Analytics Server Control" on page 89

Server Control Names	Description	More Information
Blog	<p>Ektron provides eight server controls that relate to displaying a Blog on a Web site. They are:</p> <ul style="list-style-type: none"> • The Blog server control • The BlogCalendar server control • The BlogCategories server control • The BlogEntries server control • The BlogPost server control • The BlogRecentPost server control • The BlogRoll server control • The BlogRSS server control 	<p>"Blog Server Controls" on page 43</p> <p>"Blog Server Control" on page 43</p> <p>"BlogEntries Server Control" on page 45</p> <p>"BlogPost Server Control" on page 48</p> <p>"BlogCalendar Server Control" on page 50</p> <p>"BlogRoll Server Control" on page 54</p> <p>"BlogCategories Server Control" on page 55</p> <p>"BlogRecentPosts Server Control" on page 56</p> <p>"BlogRSS Server Control" on page 57</p>
BreadCrumb	Creates a breadcrumb trail of where site visitors have been and allows them to navigate back to previous pages.	<p>"BreadCrumb Server Control" on page 59</p> <p>For information on other BreadCrumb types, see "FolderBreadcrumb Server Control" on page 69 and "SiteMap Server Control" on page 235</p>
BusinessRules	Adds a Business Rule created in the CMS400.NET Workarea to a Web form.	"BusinessRules Server Control" on page 73
Calendar	Displays an event calendar.	"Calendar Server Control" on page 75
Collection	Displays links generated by collections.	"Collection Server Control" on page 82
ContentBlock	Displays and manages a content block.	"ContentBlock Server Control" on page 91
ContentList	Displays a list of content blocks created from a comma delimited list.	"ContentList Server Control" on page 100
ContentRating	Place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item.	"ContentRating Server Control" on page 110

Server Control Names	Description	More Information
DesignTimeDiagnostic	The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service.	"DesignTimeDiagnostic Server Control" on page 115
DhtmlMenu	Displays a drop down menu with submenus.	"DhtmlMenu Server Control" on page 160
FolderBreadcrumb	FolderBreadcrumbs show you the path through the sitemap to the current page.	"FolderBreadcrumb Server Control" on page 69 For information on other Breadcrumb types, see "Breadcrumb Server Control" on page 59 and "SiteMap Server Control" on page 235
FormBlock	Displays a content block that is associated with a form.	"FormBlock Server Control" on page 117
Forum	Forums provide a discussion board where topics can be discussed on your Web site.	"Forum Server Control" on page 120
HTMLEditor	The server control displays the Ektron CMS400 HTML editor on a Web page. By adding the HTMLEditor to a Web page, site visitors can create or edit HTML content for a site or project.	"HTMLEditor Server Control" on page 122
IndexSearch	Perform XML index searches on XML content. (XML content is created using Smart Forms.)	"IndexSearch Server Control" on page 218 For information on searching other content types, see "Search Server Control" on page 182
LanguageAPI	Allows developers to force a language for a Web site.	"LanguageAPI Server Control" on page 130
LanguageSelect	Lets user select language of site.	"LanguageSelect Server Control" on page 133
ListSummary	Displays a list of content block summaries.	"ListSummary Server Control" on page 135

Server Control Names	Description	More Information
Login	Displays the login or logout button, which lets the user log in and use Ektron CMS400.NET.	"Login Server Control" on page 143
Membership	Creates a form for site visitors to add themselves to your Membership users list.	"Membership Server Control" on page 146
Menu	Calls a defined menu to be used with an Xslt to display the menu.	"Menu Server Control" on page 152
MetaData	Adds metadata contained in content blocks to a Web page.	"MetaData Server Control" on page 167
MetaDataList	Creates a list of content blocks based on the metadata contained in the content blocks.	"MetaDataList Server Control" on page 171
Poll	Displays a poll or survey created from an Ektron CMS400.NET form on a Web page.	"Poll Server Control" on page 176
RssAggregator	Aggregates an RSS feed from an RSS Web site.	"RssAggregator Server Control" on page 179
Search	Inserts a text box with a search button in the template. The site visitor enters criteria and launches a search of HTML content, forms, and assets on your Web site.	"Search Server Control" on page 182 For information on using XML Indexing search, see "IndexSearch Server Control" on page 218
Sitemap	The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site.	"SiteMap Server Control" on page 235 For information on other Breadcrumb types, see "Breadcrumb Server Control" on page 59 and "FolderBreadcrumb Server Control" on page 69

Blog Server Controls

IMPORTANT!

In addition to reading this section on the Blog server controls, reading the User Manual section "Blogs" is recommend.

Ektron provides eight server controls that relate to displaying a Blog on a Web site. One of these, the Blog server control, allows you to display all of the pieces that typically make up a blog site on a form. These items are blog entries, blog posts, blog categories, a blog's recent posts, a blog's RSS feed, a blog calendar and a blog roll. The other seven Blog server controls allow you to display each of these items individually. The seven individual Blog server controls offer additional ways to customize the page layout and appearance of the items.

The eight Blog server controls provided by Ektron are

- "Blog Server Control" on page 43
- "BlogEntries Server Control" on page 45
- "BlogPost Server Control" on page 48
- "BlogCalendar Server Control" on page 50
- "BlogRoll Server Control" on page 54
- "BlogCategories Server Control" on page 55
- "BlogRecentPosts Server Control" on page 56
- "BlogRSS Server Control" on page 57

Using the Blog Server Control vs. Using the Individual Blog Server Controls

When you use the Blog server control to display a blog, you only have to deal with one server control and its settings. This makes adding a blog to your site quick and easy.

Using the Individual Blog server controls to display a Blog, allows you the flexibility to display the parts of a Blog you want to display. For example, you might to display the Blog entries and a Blog roll, but not display a Blog calendar. In addition, using the individual Blog server controls allows you to further define the display details of each control.

Blog Server Control

The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control. These items include blog posts, a comments link, a blog roll, blog categories, recent blog posts, the RSS feed for a blog and a blog calendar. If you want to learn more about each item, see the User Manual section on "Blogs"

Using the Blog server control allows you to easily maintain the overall look of the blog. When you change display properties, such as the background color or the font, they change for the whole blog not just a piece of it.

While you can change display properties, you can not change the location of each item in the server control. The title always appears on the top, with the tagline below it. The blog posts always appear to the left and the calendar and blog roll to the right.

The Story of My Life
This is the Story of My Life and I Write it Everyday

02/14/2006

 **Writing a Masterpiece** (Music)

*I'm gonna write the melody
That's gonna make history
Yeah, and when I paint my masterpiece
I swear I'll show you first
There just ain't a way to see
Who, when, why or what will be
'Til now is then it's a mystery
A blessing and a curse
Or something worse, yeah*

Posted by John Edit at 02/14/2006 10:37:55 AM | [Comments \(0\)](#)

Jan February Mar
2006

Jan	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

Blogroll

[Scott's Blog](#)
[Toby Tyler's Blog](#)
[Music Artist of the Week](#)
[Best Week Ever](#)


The Blog server control is easy to use. Simply add it a Web form and set the BlogPath property to the folder location in the CMS400.NET. Next, choose whether to enable RSS and whether to show the header and tagline.

If you want to further customize the look of the blog, you can edit the Blog.css file located in webroot\CMS400Demo\Workarea\csslib.

Blog Server Control Properties

The following table explains the properties of the Blog control.

Note	The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.
------	--

Property	Value	Data Type
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .	Integer
ShowHeader	Shows the title and tagline when set to <code>True</code> . The default is True . True - show title and tagline. False - do not show header and tagline.	Boolean
ShowRSS	Displays the icon for the RSS feed () when set to <code>True</code> . The default is True . True - show RSS feed icon. False - do not show RSS feed icon.	Boolean

BlogEntries Server Control

The BlogEntries server control allows you to display a list of multiple blog posts on a Web form. By using this server control, you can add a blog posts to a Web form without having to add items normally contained in a blog.

Using this server control in combination with the other individual blog server controls, allows you the flexibility to change the appearance of this control without changing other blog controls. In addition, you can layout the controls any way you like, unlike the Blog server control. Below is an example of a BlogEntries server control.

02/14/2006

Writing a Masterpiece (Music)

I'm gonna write the melody

That's gonna make history

Yeah, and when I paint my masterpiece

I swear I'll show you first

Posted by John Edit at 02/14/2006 10:37:55 AM | [Comments \(0\)](#)

The Story of My Life

This is the story of my life

And I write it everyday

I know it isn't black and white

And it's anything but gray

Posted by Application Administrator at 02/14/2006 10:35:34 AM | [Comments \(2\)](#)

You can use the BlogCalendar and BlogEntries server controls together to allow blog readers to choose which day's blog posts they see. For information on using these server controls together, see ["Using the BlogCalendar and BlogEntries Server Controls Together"](#) on page 51.

BlogEntries Server Control Properties

The following table explains the properties of the BlogEntries control.

Note [The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.](#)

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String

Property	Value	Data Type
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
BlogPostParameter	Works like the <code>DynamicParameter</code> for content blocks. When <code>id</code> is selected, this server control passes the blog post ID as a URL parameter. If you do not set this parameter to <code>id</code> , you will not be forwarded to the blog post's page when you click on any links in the post. The default setting is <code>id</code> . Blank - the list of blog posts is static. The links in the blog posts are inactive. id - the id of the blog post is passed to the URL as a parameter and None use default - the list of blog posts is static. The links in the blog posts are inactive.	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .	Integer
DateToStart	The date of the last blog entries you want to appear. For example, if you want to display blog entries for January 1, 2006 and before, you enter 1/1/2006. Clicking the dropdown box provides you with a calendar.	DateTime
DisplayXslt	Specify an external XSLT file. See Also: "The OverrideXslt and DisplayXslt Properties" on page 96 .	String
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer

Property	Value	Data Type
MaxResults	The maximum number of entries displayed. 0 (zero) = the default set in the Workarea blog property # Post Visible. The default value is 0 (zero).	Integer
ShowHeader	Shows the title and tagline when set to True. The default is True . True - show title and tagline. False - do not show header and tagline.	Boolean
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

BlogPost Server Control

The BlogPost server control allows you to display an individual blog post on a page.

There are two ways this server control displays a blog post. First, if a user is logged in as a CMS400.NET user or a Membership user, the control displays the blog post, comments and comments form on the web page. The second way is if a user is not logged in, the control displays only the blog post.

NOTE the Enable Comments property in the workarea must be enabled for comments and the comments form to appear.

BlogPost Server Control Properties

The following table explains the properties of the BlogPost control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String

Property	Value	Data Type
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.	Integer
DefaultContentID	The ID of a blog post that appears where you inserted this server control if no other content block is identified, or is not available. If you don't know the ID number of the blog post, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DisplayXslt	Specify an external XSLT file. See Also: "The OverrideXslt and DisplayXslt Properties" on page 96.	String
DynamicParameter	To make this blog post dynamic, select id . When you do, this server control uses the blog post passed as a URL parameter.	String
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
OverrideXslt	An internal Xslt file used to display XML content if one is not identified in the Xslt property. See Also: "The OverrideXslt and DisplayXslt Properties" on page 96	must be one of these values: default, one, two, three
ShowType	This property is for future use.	
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

BlogCalendar Server Control

The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog. When a Blog calendar is associated with a blog, the days that have blog posts are highlighted on the calendar.

You can use a BlogCalendar server control with a BlogEntries server control. This gives site visitors a way to display blog posts for a given day. For an example of using the BlogCalendar and BlogEntries server controls together, see ["Using the BlogCalendar and BlogEntries Server Controls Together"](#) on page 51

BlogCalendar Server Control Properties

The following table explains the properties of the BlogCalendar control.

Note [The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.](#)

Property	Value	Data Type
Authenticated	<p>Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p>	String
BlogID	<p>The ID of the blog in CMS400.NET. For example: 41.</p> <p>If you don't know the ID of the blog, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p>	String

Property	Value	Data Type
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: " Caching with Server Controls " on page 36.	Integer
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

Using the BlogCalendar and BlogEntries Server Controls Together

You can use the BlogCalendar and BlogEntries server controls together to allow blog readers to choose which day's blog posts they see.

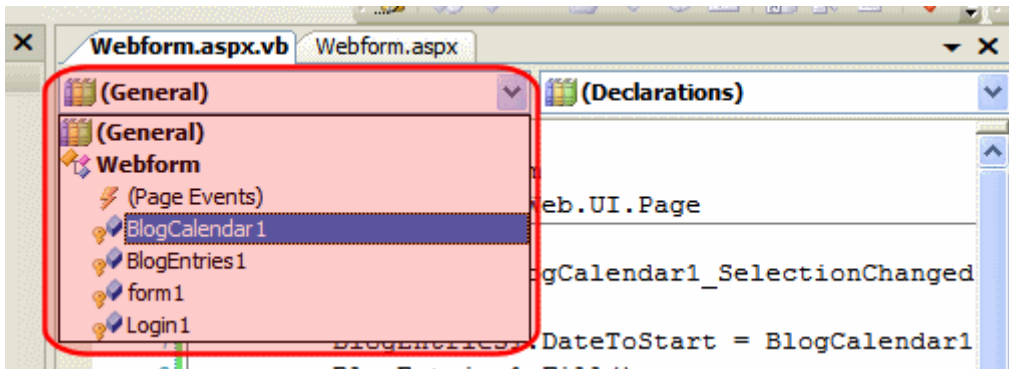
Steps to Use BlogCalendar and BlogEntries Together

1. On a Visual Studio Web form, add a BlogCalendar server control.
2. In the Ektron properties, set the BlogPath to the blog you want to use.

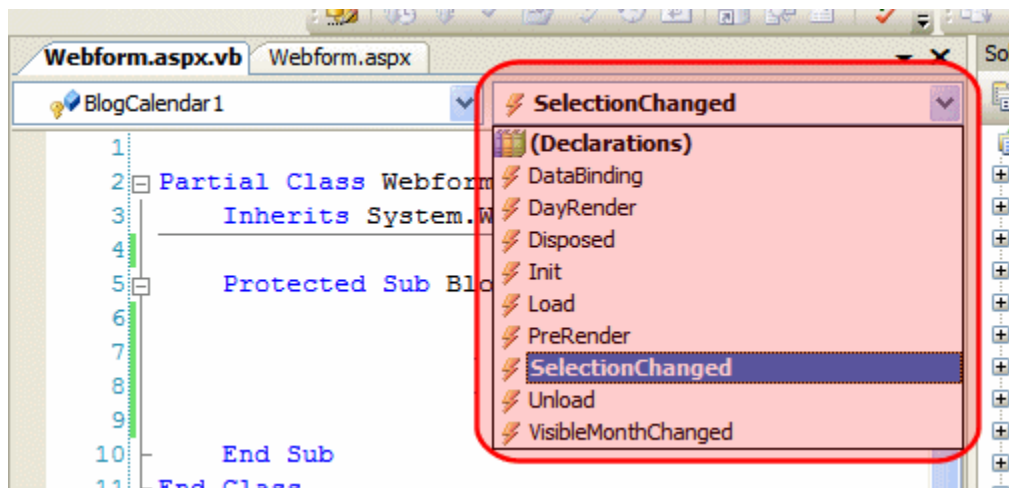


3. If you want, change the appearance properties for the calendar. (This step is optional.)
4. Add a BlogEntries server control.
5. If you want, change the appearance properties for the BlogEntries server control. (This step is optional.)

6. On the code behind page, set the **Class Name** list to **BlogCalendar1**.



7. Set the **Method Name** list to **SelectionChanged**.



8. In the Protected Sub **BlogCalendar1_SelectionChanged**, Add the following line of code.

```
BlogEntries1.DateToStart = BlogCalendar1.SelectedDate
```

NOTE This line of code equates the **DateToStart** property for the **BlogEntry** control to the **SelectedDate** property of the **BlogCalendar** control.

```

1
2 Partial Class Webform
3     Inherits System.Web.UI.Page
4
5     Protected Sub BlogCalendar1_SelectionChanged(ByVal sender As Object,
6         ByVal e As System.EventArgs) Handles BlogCalendar1.SelectionChanged
7
8         BlogEntries1.DateToStart = BlogCalendar1.SelectedDate
9
10    End Sub
11 End Class

```

9. Click the **Build Menu > Build Page**.
10. When you view page in a browser and click on a date in the calendar that has blog posts, you see the posts for that day.

02/14/2006

Writing a Masterpiece (Music)

I'm gonna write the melody
That's gonna make history
Yeah, and when I paint my masterpiece
I swear I'll show you first

Posted by John Edit at 02/14/2006 10:37:55 AM | [Comments \(0\)](#)

The Story of My Life

This is the story of my life
And I write it everyday
I know it isn't black and white
And it's anything but gray

Posted by Application Administrator at 02/14/2006 10:35:34 AM | [Comments \(2\)](#)

February 2006						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

BlogRoll Server Control

The BlogRoll server control displays the blog roll for a blog on a Web form. Blog rolls are created in the Workarea for each blog. This control reads that list and displays a clickable list of links. For more information on the blog roll, see the User Manual Section "Blogs" > "Blog Properties" > "The Blog Roll".

Typically this control is displayed along side other Individual Blog server controls. Using the individual server controls, such as the Blog Roll server control, allows you to have more flexibility for layout and appearance than if you use Blog server control.

BlogRoll Server Control Properties

The following table explains the properties of the BlogRoll control.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .	Integer
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer

Property	Value	Data Type
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

BlogCategories Server Control

The BlogCategories server control displays the blog categories for a blog on a Web form. Blog categories are created in the Workarea for each blog. This control reads that list and displays a clickable list of links. When a link is clicked it displays all the posts associated with the category.

Typically this control is displayed along side other Individual Blog server controls. Using the individual server controls, such as the BlogCategories server control, allows you to have more flexibility for layout and appearance than if you use Blog server control.

BlogCategories Server Control Properties

The following table explains the properties of the BlogCategories control.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String

Property	Value	Data Type
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.	Integer
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

BlogRecentPosts Server Control

The BlogRecentPosts server control displays a list of recent blog posts links on a Web form. when a link is clicked it takes you to the blog post. You can control how many recent post links are displayed by setting the `NumberOfPosts` property.

Typically this control is displayed along side other Individual Blog server controls. Using the individual server controls, such as the BlogRecentPosts server control, allows you to have more flexibility for layout and appearance than if you use Blog server control.


BlogRecentPosts Server Control Properties

The following table explains the properties of the BlogRecentPosts control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .	Integer
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
NumberOfPosts	Sets the number of post links to display.	Integer
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

BlogRSS Server Control

The BlogRSS server control displays the icon () for the blog's RSS feed on the Web form. When the icon is clicked, the RSS feed for the blog appears.

Typically this control is displayed along side other individual Blog server controls. Using the individual server controls, such as the BlogRSS server control, allows

you to have more flexibility for layout and appearance than if you use Blog server control.

BlogRSS Server Control Properties

The following table explains the properties of the BlogRSS control.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String
BlogID	The ID of the blog in CMS400.NET. For example: 41. If you don't know the ID of the blog, use the CMS Explorer to browse to it. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See <i>Also: "Caching with Server Controls" on page 36.</i>	Integer
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

BreadCrumb Server Control

There are two kinds of breadcrumbs: sitemap breadcrumbs and history breadcrumbs.



Sitemap breadcrumbs show you the path through the sitemap to the current page, like the path shown at the top of the page of a Yahoo! directory listing. Sitemap breadcrumbs always show the same path for the same page, regardless of how that page is reached. To use a sitemap breadcrumb, see "[FolderBreadcrumb Server Control](#)" on page 69.



History breadcrumbs more accurately fulfill the breadcrumb trail metaphor. They display an edited version of the browser's history - links to each of the most recently visited pages in the order that they were visited.

The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.

The breadcrumb trail can appear vertically or horizontally on a Web page. In addition, you can define how many breadcrumbs are left for site visitors to navigate back. You can also use an image and Alt text instead of, or in addition to, text to display the breadcrumb trail.

NOTE If a site visitor revisits a page that is in the breadcrumb trail already, the breadcrumb trail reverts back to that point in the trail. For example, if you visit the following pages Dev > FAQ > WebForm > PR, and then return to FAQ, the breadcrumb trail looks like this: **Dev > FAQ**.

Breadcrumb Trail Type	Example
Horizontal; text only	Dev > WebForm > pr > FAQ > Meta*
Horizontal with images	
Horizontal with images and Alt text	

Breadcrumb Trail Type	Example
Vertical; text only	<pre> > Dev > WebForm > pr > FAQ > Meta* </pre>
Vertical with images	 <p>A vertical breadcrumb trail where each item is enclosed in a blue rectangular button with rounded corners. The items are 'DEV', 'PR', 'FAQ', 'Search', and 'Meta', each followed by a right-pointing chevron '>'.</p>
Vertical with images and Alt text	 <p>A vertical breadcrumb trail similar to the previous one, but with an additional feature. A mouse cursor is shown hovering over the 'Meta' button, which has triggered a small tooltip box containing the word 'Meta'.</p>

You should add a BreadCrumb server control to every Web form in your site. Doing this ensures that a breadcrumb trail is created for each Web page.

This subsection contains the following topics:

- ["BreadCrumb Properties Apply to Web Forms, Not Web Pages" on page 61](#)
- ["Types of BreadCrumb Server Control Properties" on page 61](#)
- ["BreadCrumb Server Control Properties" on page 62](#)
- ["Using the BreadCrumb Server Control" on page 66](#)
- ["Making a Content Block's Title Appear in the Breadcrumb Trail" on page 67](#)

BreadCrumb Properties Apply to Web Forms, Not Web Pages

Keep in mind that your Web site is made up of Web forms. Each page depends on a form to determine much of its appearance. Forms and pages have a parent > child relationship. That is, any form can be associated with any number of pages.

When defining breadcrumb properties, you define them for a *form*. All pages that use that form inherit its breadcrumb properties.

Assume, for example, that a Web form is used for the Human Resources section of your Web site. You might use the title **Human Resources** to identify the form in the BreadCrumb server control. Whenever a page is visited that uses that form, **Human Resources** appears on the breadcrumb trail -- that is, the form title appears, not the individual page.

Types of BreadCrumb Server Control Properties

BreadCrumb server control properties are divided into two categories:

- ["Properties that Determine a Breadcrumb Trail's Appearance" on page 61](#)
- ["Properties that Determine How Pages Associated with the Form Appear on a Breadcrumb Trail" on page 62](#)

Properties that Determine a Breadcrumb Trail's Appearance

For each Web form, you can customize the breadcrumb trail. For example, you might want the breadcrumb trail to appear horizontally on one Web form and vertically on another.

Property	Determines this aspect of the breadcrumb trail's appearance on this form	For more information, see
CurrentPageIndicator	Symbols or characters to identify the current page	"CurrentPageIndicator" on page 63
DisplayStyle	Whether it appears horizontally or vertically	"DisplayStyle" on page 63
LinkLastItem	Whether the last item is a hyperlink	"LinkLastItem" on page 64
LinkTarget	The type of window that appears when a user clicks an item	"LinkTarget" on page 64
MaxItems	The maximum number of items	"MaxItems" on page 65

Property	Determines this aspect of the breadcrumb trail's appearance on this form	For more information, see
Mode	whether the breadcrumb trail appears as hyperlinked text or plain text.	"Mode" on page 65
Separator	Symbols or characters used to separate items	"Separator" on page 65

Properties that Determine How Pages Associated with the Form Appear on a Breadcrumb Trail

Use these properties define how any page that uses this form appears within a breadcrumb trail. It does not matter where the trail appears.


Note that you can use text, an image, or both to describe the form within the breadcrumb trail. If you use both, the image appears first, followed by the text.

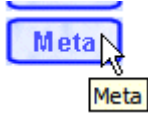
Property	Determines this aspect of Web form when it appears on breadcrumb trail	For more information, see
DisplayTitle	Text to describe it	"DisplayTitle" on page 63
IconAlt	"Alt" text associated with image specified in IconPath property	"IconAlt" on page 64
IconPath	Image to define it	"IconPath" on page 64

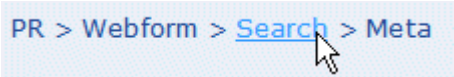
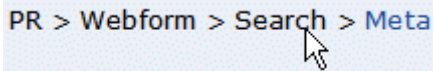

BreadCrumb Server Control Properties

The BreadCrumb server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.


Property	Response	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CurrentPageIndicator	<p>If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail. See example below.</p> <p style="text-align: center;">Dev > WebForm > pr > FAQ > Meta </p> <p>These characters appear <i>after</i> the image or title that identifies the current page in the breadcrumb trail.</p>	String
DefaultContentID	Gets or Sets the ContentID for the BreadCrumb display title.	Integer
DisplayStyle	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal. See Also: "Breadcrumb Trail Type" on page 59	String
DisplayTitle	<p>Enter text to describe this Web form when it appears in the breadcrumb trail. For example, if the Web form whose properties you are defining is used for all Human Resources pages on your site, enter Human Resources. See Also: "BreadCrumb Properties Apply to Web Forms, Not Web Pages" on page 61</p> <p>The default is Title.</p> <p>If you define an image in the IconPath property below, the image appears in the trail, followed by this text.</p>	String
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.	String
Hide	<p>Used to hide the breadcrumb trail in design time and run time.</p> <p>True = Hide breadcrumb trail</p> <p>False = Show breadcrumb trail</p>	Boolean

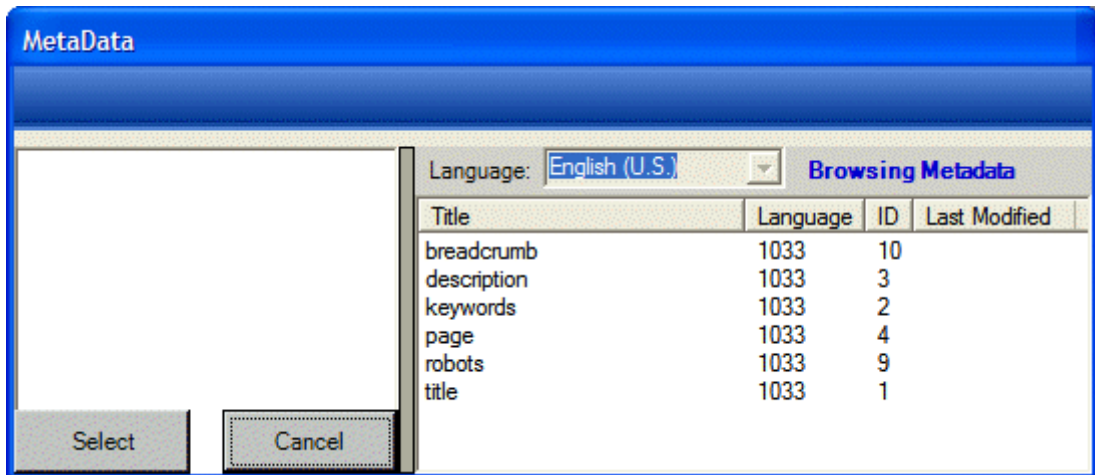
Property	Response	Data Type
IconAlt	<p>If you define an image in the IconPath property, enter any "Alt" text that should appear when a site visitor hovers the cursor over that image. Here is an example.</p> 	String
IconPath	<p>If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image.</p> <hr/> <p>Important: The image location must be relative to the Web root.</p> <hr/> <p>For example: \cms400demo\Workarea\Images\bc_meta_icon.gif</p> <p>On the breadcrumb trail, the image precedes any text defined in the DisplayTitle property.</p>	String
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
LinkLastItem	<p>Use this property to determine whether the last breadcrumb item appears as a hyperlink on this Web form. If this property is set to true, and a user clicks the item, the current page reappears.</p> <p>True = last item is a hyperlink False = last item is an image and/or text only; the user cannot click on it</p>	Boolean
LinkTarget	<p>Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.</p> <p>_Self - opens in same window _Top - opens in parent window _Blank - opens in new window _Parent - opens in the parent frame</p>	String

Property	Response	Data Type
MaxItems	<p>Enter the maximum number of items in the breadcrumb trail on this Web form.</p> <p>The default is 5.</p> <p>If you set a value of 1 or greater and the user navigates beyond that number of pages, only the most recent pages appear. The older pages disappear from the trail.</p> <p>To allows an unlimited number of breadcrumb trail items on this Web form, enter zero (0).</p>	Integer
MetadataName	<p>Specify the name of a Metadata Type that you want to associate with the page. For more information, see "BreadCrumb Metadata Type" on page 65</p>	String
Mode	<p>Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.</p> <p>Normal = breadcrumb trail is hyperlinked</p>  <p>DisplayOnly = breadcrumb trail is plain text</p> 	String
Separator	<p>Enter one or more characters to separate the items in a breadcrumb trail on this Web form.</p> <p>The default character is the greater than sign (>).</p> 	String
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

BreadCrumb Metadata Type

To associate a Metadata Type with the BreadCrumb server control, follow these steps.

1. In the properties window for the BreadCrumb server control, click on the MetadataName property.
2. Click the Ellipses button ().
If you are not logged in, log in now using the CMS Explorer window.
3. The Browsing Metadata screen appears.



4. Select a Metadata Type to apply to the BreadCrumb server control.
5. Set the DynamicParameter to ID. This allows the metadata to reflect what ever content block appears on the page.

Using the BreadCrumb Server Control

Add the BreadCrumb server control to each Web form for which you want to create a breadcrumb trail. Follow these steps to use the BreadCrumb server control.

WARNING!

It is recommended that you do not add a BreadCrumb server control to a Web form that contains a Calendar server control. When both server controls appear on the same page and a site visitor clicks on different months, the BreadCrumb trail could look like this: Calendar > Calendar > Calendar > Calendar > Calendar. This happens because each time a site visitor clicks on a month, you are in effect opening a new Web form.

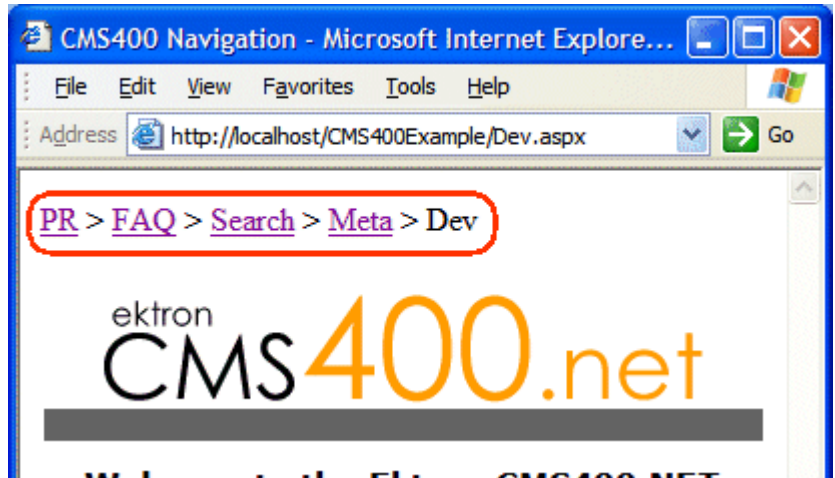
1. Open a Web form for which you want to create a breadcrumb trail.
2. Drag and drop the BreadCrumb server control onto an appropriate location of the Web form.
3. Set the **DisplayTitle** property. See "[DisplayTitle](#)" on page 63.

NOTE

To customize the look of your breadcrumb trail, configure the other properties. For more information, see "[BreadCrumb Server Control Properties](#)" on page 62.

4. Save the Web form.

5. Repeat steps one through four for all Web forms to be included in the breadcrumb trail.
6. Open a browser.
7. View a Web page with a BreadCrumb server control in it.
8. Navigate to the rest of the Web pages that contain BreadCrumb server controls.
9. As you move around, the pages are added to the breadcrumb trail.



Making a Content Block's Title Appear in the Breadcrumb Trail

By adding a line or two of code, the breadcrumb trail can display the title of a content block. You can add a line of code to each Web form or if your content is dynamic, add the code once and each time a new piece of content is called dynamically a new breadcrumb is created.

Only one content block per Web form can be used with the breadcrumb. If you have multiple content blocks on a page, select the one that describes the overall page best.

Steps to Use a Content Block's Title as a Breadcrumb

1. Drag and drop a **ContentBlock** server control on a Web form.
2. Choose a **DefaultContentID** for the content block.
3. Drag and drop a **BreadCrumb** server control on the same form.
4. Remove the word **Title** from the **DisplayTitle** property.
5. Add the following line code to the page load event code behind:

```
BreadCrumb1.DisplayTitle = ContentBlock1.EkItem.Title
```
6. Build the project.
7. View the Web form in a Web browser.

The following is an example content block titles used as breadcrumbs:



The word Title appears in the breadcrumb trail when the DisplayTitle property is left empty. Even with the added code, if the breadcrumb loads before the content block, the breadcrumb has no information in the DisplayTitle property and therefore displays the word title by default.

If the word Title appears in the breadcrumb trail, check to make sure the word Title does not appear in the DisplayTitle property. Next, if Title still appears, add the following line of code to the page load event of the code behind:

```
ContentBlock1.Fill()
```

The code in the page load event should now look like this:

```
ContentBlock1.Fill()  
BreadCrumb1.DisplayTitle = ContentBlock1.EkItem.Title
```

This ensures that the content block information is loaded first.

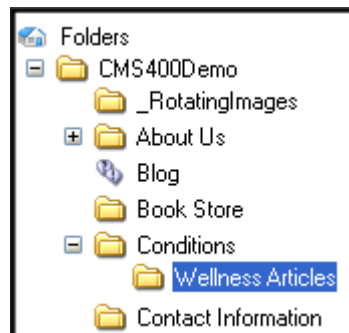
FolderBreadcrumb Server Control

As mentioned in the BreadCrumb server control section, there are two kinds of breadcrumbs: sitemap breadcrumbs and history breadcrumbs. Ektron handles history breadcrumbs with the BreadCrumb server control. See *Also*: "[BreadCrumb Server Control](#)" on page 59.

Sitemap breadcrumbs show you the path through the sitemap to the current page, like the path shown at the top of the page of a Yahoo! directory listing. Sitemap breadcrumbs always show the same path for the same page, regardless of how that page is reached.

The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. For example, the FolderBreadcrumb below matches the folder structure where the content is located.

Home >> Conditions >> Wellness Articles



The sitemap for each FolderBreadcrumb is set on the BreadCrumb tab located in the Workarea folder properties. See *Also*: The Administrator Manual > "Managing Content Folders" > "Breadcrumbs".

This subsection contains the following topics:

- "[BreadCrumb Server Control Properties](#)" on page 69
- "[Using the BreadCrumb Server Control](#)" on page 71

BreadCrumb Server Control Properties

The BreadCrumb server control properties are described in this table.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Response	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
DefaultContentID	Gets or Sets the ContentID for the BreadCrumb display title. The breadcrumb information is taken from the folder properties in which the content resides.	Integer
DefaultFolderID	The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID is given then it overrides this property.	Integer
DisplayStyle	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.	String
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.	String
Hide	Used to hide the breadcrumb trail in design time and run time. True = Hide breadcrumb trail False = Show breadcrumb trail	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
LinkTarget	Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self . _Self -opens in same window _Top - opens in parent window _Blank - opens in new window _Parent - opens in the parent frame	String
Mode	Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal . Normal = breadcrumb trail is hyperlinked DisplayOnly = breadcrumb trail is plain text	String

Property	Response	Data Type
Separator	Enter one or more characters to separate the items in a breadcrumb trail on this Web form. The default character is the greater than sign (>).	String
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Using the BreadCrumb Server Control

Add the FolderBreadcrumb server control to each Web form for which you want to create a breadcrumb sitemap. Follow these steps to use the FolderBreadcrumb server control.

1. Open a Web form for which you want to create a folder breadcrumb map.
2. Drag and drop the FolderBreadcrumb server control onto an appropriate location of the Web form.
3. Set the **DefaultContentID** or **DefaultFolderID** property.
4. Save the Web form.
5. Open a browser.
6. View a Web page with the FolderBreadCrumb server control in it.

7. The FolderBreadcrumb now appears on your site as a sitemap breadcrumb.



BusinessRules Server Control

The BusinessRules server control allows you to add a Business Rule created in the Workarea to a Web form. It also allows you to add a place holder for a Business Rule. Then, a CMS400.NET administrator can create a business rule at a later date.

For an explanation of Business Rules and how they work, see the Administrator Manual > "Business Rules".

BusinessRules Server Control Properties

The following table explains the properties of the Business Rules server control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. <i>See Also:</i> " Using CMS Explorer to Browse Your Ektron CMS400.NET Site " on page 15.	String
Hide	Used to hide a form block in design time and run time. True = Hide Business Rule False = Show Business Rule	Boolean
Language	Set a language for viewing form content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
RuleID	The ID of the Ruleset to evaluate. If you don't know the ID number of the Ruleset, use the CMS Explorer to browse to it. <i>See Also:</i> " Using CMS Explorer to Browse Your Ektron CMS400.NET Site " on page 15	Integer

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String
























































































































































































































































































Calendar Server Control

The Calendar server control displays a calendar that was created in the Workarea. By using the Calendar server control's properties, you can select multiple views of the calendar to display. See Also: ["DisplayType" on page 78](#)

You can adjust a calendar's look and feel by using a style sheet. See Also: ["Calendar Style Sheet" on page 80](#) and ["calendarStyles.css" on page 616](#).

The Calendar server control can output the raw XML data of the calendar. You can use this XML data to create a completely customized calendar.

Below is a Calendar displayed on the Web Page. When logged in, you can add and view calendar events. When logged out, calendar events only appear.

Login Status	Calendar display																																										
Logged in	<div><div>vember</div><div>December 2004</div><div>Januai</div><table><tr><th>Sun</th><th>Mon</th><th>Tue</th><th>Wed</th><th>Thu</th><th>Fri</th><th>Sat</th></tr><tr><td></td><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td></td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr><tr><td></td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr><tr><td></td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr><tr><td></td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td></td></tr></table></div>	Sun	Mon	Tue	Wed	Thu	Fri	Sat				1   	2   	3   	4   	  	6   	7   	8   	9   	10   	11   	  	13   	14   	15   	16   	17   	18   	  	20   	21   	22   	23   	24   	25   	  	27   	28   	29   	30   	31   	
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																					
			1   	2   	3   	4   																																					
  	6   	7   	8   	9   	10   	11   																																					
  	13   	14   	15   	16   	17   	18   																																					
  	20   	21   	22   	23   	24   	25   																																					
  	27   	28   	29   	30   	31   																																						

Login Status	Calendar display																																										
Logged Out	<div><div>November</div><div>December 2004</div><div>January</div><table><tr><th>Sun</th><th>Mon</th><th>Tue</th><th>Wed</th><th>Thu</th><th>Fri</th><th>Sat</th></tr><tr><td></td><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td></td></tr><tr><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td></td></tr><tr><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td></td></tr><tr><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td></td><td></td></tr></table></div>	Sun	Mon	Tue	Wed	Thu	Fri	Sat				1	2	3	4	6	7	8	9	10	11		13	14	15	16	17	18		20	21	22	23	24	25		27	28	29	30	31		
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																					
			1	2	3	4																																					
6	7	8	9	10	11																																						
13	14	15	16	17	18																																						
20	21	22	23	24	25																																						
27	28	29	30	31																																							

A calendar can be *static* or *dynamic*.

- A static calendar is selected in the server control properties.
- A dynamic calendar is passed to the page by a URL parameter.

This subsection contains the following topics:

- ["Calendar Server Control Properties" on page 76](#)
- ["Calendar Style Sheet" on page 80](#)


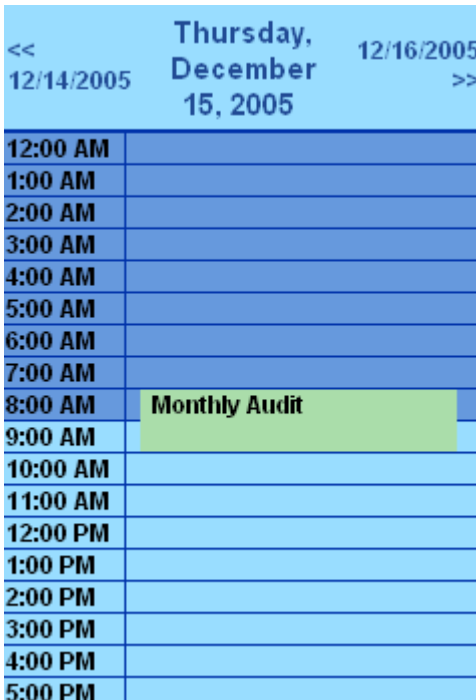
Calendar Server Control Properties

The Calendar server control properties are described in this table.

NOTE There are additional settings to help define the calendar in the Ektron CMS400.NET Workarea. These include display weekends, forward only, show all label, etc. For information on these properties, see Ektron CMS400.NET [Administrator's Manual > Calendars > Managing Calendar Modules > Viewing Calendar Modules](#).

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String
DefaultCalendarID	The ID number of the calendar that appears where you inserted this server control. If you don't know the ID number of the calendar, use the CMS Explorer to browse to it. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	Integer

Property	Value	Data Type
DisplayType	<p>Specify the Calendar view type. The default is Month.</p> <ul style="list-style-type: none"> Month - A 30 day calendar month view.  Day - Single day view.  	String

Property	Value	Data Type																														
DisplayType (Con't)	<ul style="list-style-type: none">Inline - Display events that occur from the start date of the calendar through the end date.<div><div>Thursday, December 01, 2005 - Saturday, December 31, 2005</div><div>12/15/2005</div><div>Monthly Audit Business Office Start:8:00 AM End:10:30 AM This is to test a recurring calendar event.</div><div>12/21/2005</div><div>Christmas Party Event Room Start:7:30 PM</div></div>MonthUpWithEvent - Displays a month calendar. Events appear to the right of the calendar when you click on a day.<div><div><div><< December, Jan >></div><div>Nov 2005</div><div>12/21/2005</div></div><div><table><tr><th>Mon</th><th>Tue</th><th>Wed</th><th>Thu</th><th>Fri</th></tr><tr><td></td><td></td><td></td><td>1</td><td>2</td></tr><tr><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr><tr><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td></tr><tr><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr></table><div>2004 .. 2005 .. 2006</div></div><div>Christmas Party Location:Event Room Start:7:30 PM End:11:00 PM Come to the Christmas p.</div></div> <div>Note: The URL parameters hevtype and evhighlight do not work. They are reserved for future use.</div>	Mon	Tue	Wed	Thu	Fri				1	2	5	6	7	8	9	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	
Mon	Tue	Wed	Thu	Fri																												
			1	2																												
5	6	7	8	9																												
12	13	14	15	16																												
19	20	21	22	23																												
26	27	28	29	30																												
DynamicParameter	<p>To make this calendar dynamic, select calendar_id. When you do, this server control uses the calendar passed as a URL parameter.</p>	String																														

Property	Value	Data Type
EndDate	Specify the end date the calendar displays. Used in conjunction with the StartDate property. For example, you have a calendar that shows only classes for the next three month semester.	Date
Hide	Used to hide output of calendar in design time and run time. True = Hide calendar False = Display calendar	Boolean
Language	Set a language for viewing calendar. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
ShowXML	At runtime, output the raw XML data instead of the calendar. With this XML data, you can apply your own XSL to create your own custom display formats. Default is False . True = Show raw XML data False = Show Calendar	Boolean
StartDate	Specify the start date the calendar displays. Used in conjunction with the EndDate property. For example, you have a calendar that shows only classes for the next three month semester.	Date
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Calendar Style Sheet

Before you insert a Calendar into your Ektron CMS400.NET template, Ektron recommends referencing the style sheet `\webroot\CMS400Demo\Workarea\csslib\calendarStyles.css` in your template. Copy the style sheet to your webroot folder then add it to your project. Next create a reference to `calendarStyles.css`. For example:

```
html>
<head>
  <title>Company Calendar</title>
```

```
<link href="calendarStyles.css" rel="stylesheet" type="text/
css" />
</head>
```

This can also be accomplished via drag and drop. If you drag and drop the `calendarStyles.css` from the Solution Explorer to your Web form, the reference is created in the HTML automatically.

For a description of the classes contained in `calendarStyles.css`, see ["calendarStyles.css" on page 616](#).

Collection Server Control

The Collection server control is used to display a list of content block links that were created as a collection in the Ektron CMS400.NET Workarea. The control is highly customizable, allowing you to define the display of the collection on a Web page.

When added to a template and visited, collections can look like this (The following illustration shows three collections on the same page of a sample site.) You can modify the display by editing its properties.

Human Resources Forms

[While you were out...](#)

Employee Forms

[Absence Request Form](#)

[Direct Deposit Form](#)

Manager Forms

[Personnel Action Form](#)

This subsection contains the following topics:

- ["Collection Server Control Properties" on page 82](#)
- ["Example of ecmNavigation Display" on page 85](#)
- ["Example of ecmTeaser Display" on page 85](#)
- ["Retrieving the XML Structure of a Collection" on page 86](#)
- ["Using the Collection Server Control Programmatically Example" on page 87](#)

Collection Server Control Properties

The Collection server control properties are described in this table.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Description	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .	Integer
ContentParameter	Checks the QueryString for this value and replaces the collection with a content block when specified. Leave blank to always display the Collection or List Summary.	String
DefaultCollectionID	The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available. If you don't know the ID number of the collection, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DisplayXslt	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every content block in the collection See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists the title of every content block in the collection plus the content summary See Also: "Example of ecmTeaser Display" on page 85 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page	String
DynamicParameter	To make this collection dynamic, select coll_id . When you do, this server control uses the collection passed as a URL parameter.	String

Property	Description	Data Type
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the collection. For example, to display content inside a web server control such as a GridView.	Boolean
Hide	Used to hide output of collection in design time and run time. True = Hide collection False = Display collection	Boolean
Language	Set a language for viewing the collection. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
LinkTarget	Defines the way a link acts when a link is clicked. Choices are: _blank - This target causes the link to always be loaded in a new blank window. This window is not named. _self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. _parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent. _top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.	String
Random	Set to True if you want to randomly display one collection item. The item changes each time a user views the page.	Boolean

Property	Description	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Example of ecmNavigation Display

The following is an example of a collection being used as a navigation menu.



ecmNavigation XSL code

Below is the XSL code that is used to create ecmNavigation Display. You can use this code as the basis to design your own XSLT.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/
XSL/Transform">
<xsl:template match="/">
<table border="0" cellspacing="0" cellpadding="0" width="100%">
<xsl:for-each select="Collection/Content">
<tr><td><a><xsl:attribute name="href">
<xsl:value-of select="QuickLink"/></xsl:attribute><xsl:value-of
select="Title"/>
</a></td></tr></xsl:for-each></table></xsl:template></
xsl:stylesheet>
```

Example of ecmTeaser Display

The following is an example of a collection using the ecmTeaser display style.

Employment Opportunities

[Plastic Molder #123](#)

RC International is looking for an experienced plastics molder.

[Servo Control Engineer #124](#)

RC International is looked for a highly skilled servo control engineer to join our team.

ecmTeaser XSL code

Below is the XSL code that is used to create ecmTeaser Display. You can use this code as the basis to design your own XSLT.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/
XSL/Transform">
<xsl:template match="/"><table border="0" cellspacing="0"
cellpadding="0" width="100%">
<xsl:for-each select="Collection/Content"><tr><td><a>
<xsl:attribute name="href"><xsl:value-of select="QuickLink"/></
xsl:attribute>
<xsl:value-of select="Title"/></a>&#160;
// xml.Append("<xsl:value-of
select="substring(DisplayDateModified, 0, string-
length(DisplayDateModified)-11)" />
</td></tr><tr><td><xsl:value-of select="Teaser" /></td></
tr><tr><td>&#160;</td></tr>
</xsl:for-each></table></xsl:template></xsl:stylesheet>
```

Retrieving the XML Structure of a Collection

Retrieving the XML structure of XML content allows for greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a Collection server control onto it.
3. Set the DefaultCollectionID property.
4. Drag and drop a Textbox on the Web form.
5. Set the TextMode property to MultiLine.

NOTE [It is also recommended that you set the width of the text box to at least 400px.](#)

6. On the code behind page, add the following line.

```
Textbox1.Text = Collection1.XmlDoc.InnerXml
```
7. Build the project.
8. View the Web form in a browser.

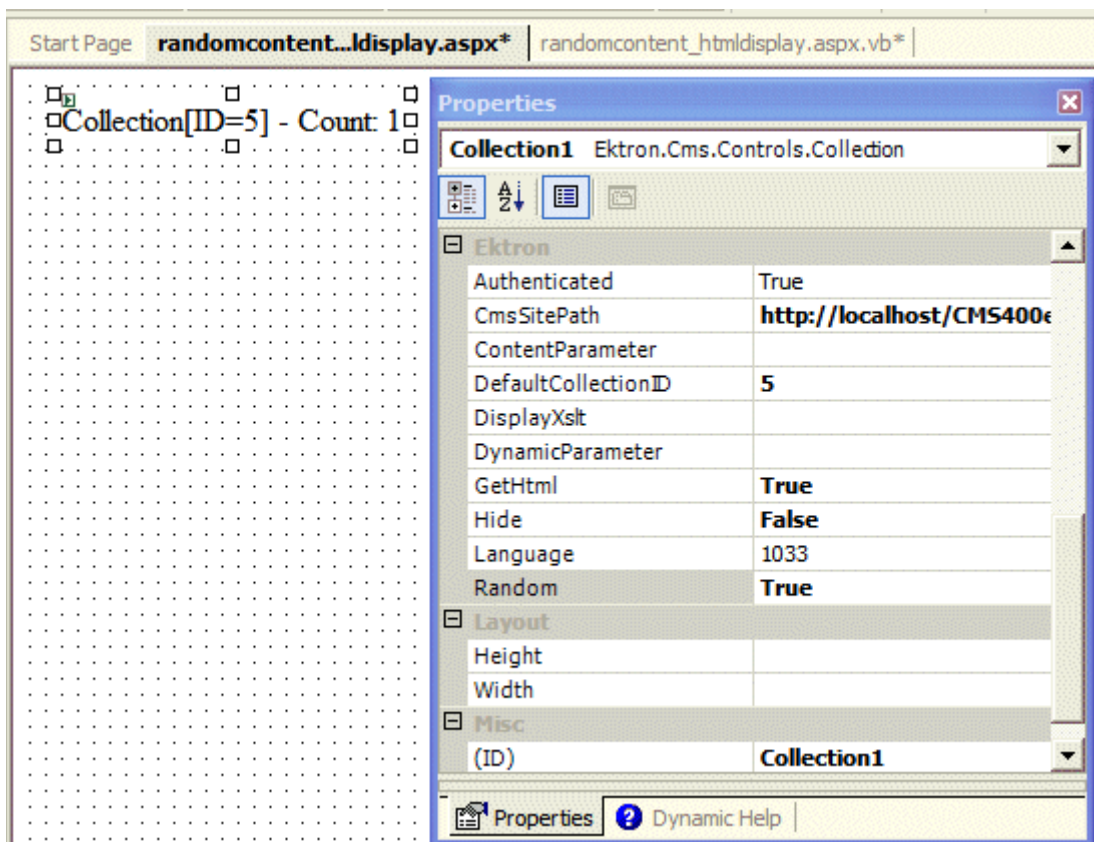
9. The XML structure of the collection appears in the textbox.

Using the Collection Server Control Programmatically Example

The following is an example of using code behind, and a drag and drop Collection server control to display random content from a collection.

1. Drag and Drop a Collection server control on your Web form.
2. Set the properties in the properties window.

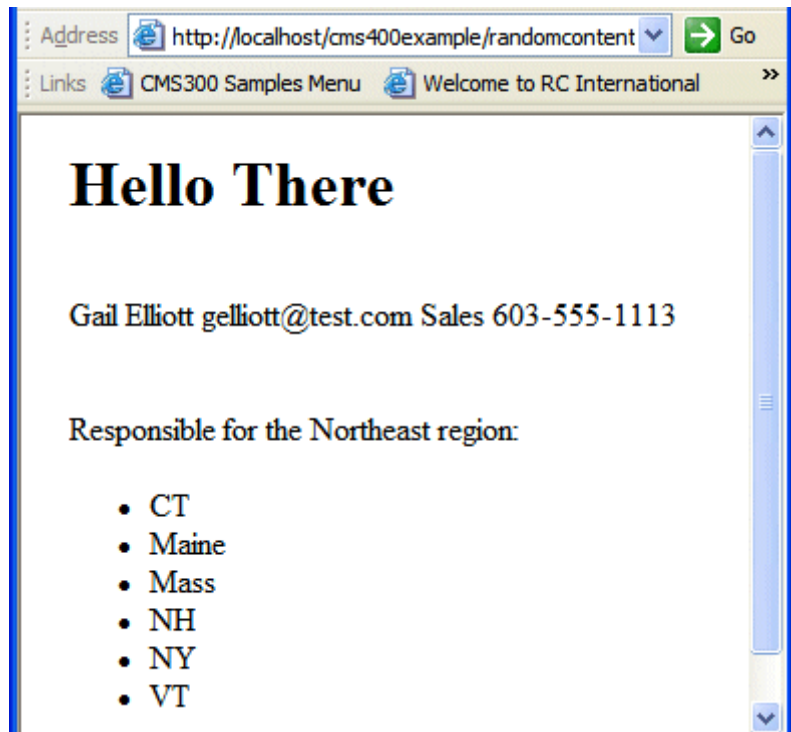
NOTE In this example, the Random property and the GetHtml property must be set to **True**.



3. Add the following code to the code behind.

```
Dim str As String
str = "<h1>Hello There</h1><br>"
str &= Collection1.EkItems(0).Html
Collection1.Text = str
```

4. Build and browse your web form.



Content Analytics Server Control

Use the AnalyticsTracker server control to track statistics about visits to your Web site, such as:

- how often your content was viewed
- how many site visitors viewed for first time, and how many returned
- the most popular referral pages (that is, the Web page from which visitors clicked to arrive on your site)

For more information about Content Analytics, see the **Ektron CMS400.NET** Administrator manual chapter "Content Analytics Feature."

Content Analytics Server Control Properties

Note [The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.](#)

NOTE

Ektron Properties

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Boolean
DefaultContentID	If desired, enter the ID of a content item being tracked by this server control. If you do, all visits to this content item are tracked for content analytic statistics. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DynamicParameter	To make this content analytic control dynamic, select id . When you do, this server control is attached to the content block passed as a URL parameter.	String

Property	Value	Data Type
Hide	Select False to display this server control on the page. Select True to suppress it.	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element. Div - The <code><div></code> tag is used when you want to apply attributes to a block of code. Custom - Lets you use custom tag.	String
EnableAnalytics	Determines if the Analytics feature is enabled for this page and (if identified) content item. ConfigSpecified - Take the value from the web.config file's <code>enableAnalytics</code> property. True - Enable Analytic tracking False - Disable Analytic tracking See Also: Ektron CMS400.NET Administrator Manual Section "Content Analytics Feature" > "Enabling the Content Analytics Feature."	

ContentBlock Server Control

The ContentBlock server control displays a content block on an Ektron CMS400.NET Web page.

Ektron CMS400.NET has two types of content blocks:

- Static - displays one specified content block
- Dynamic - displays the content block of the ID passed through a URL parameter

In addition, a content block can contain XHTML or XML content.

For an example of using the ContentBlock server control programmatically, see ["Using the ContentBlock Server Control Programmatically Example" on page 98](#)

This subsection contains the following topics:

- ["Static Content Block" on page 91](#)
- ["ContentBlock Server Control Properties" on page 91](#)
- ["Dynamic Content Block" on page 93](#)
- ["XML Content Block" on page 95](#)
- ["The OverrideXslt and DisplayXslt Properties" on page 96](#)
- ["Using the ContentBlock Server Control Programmatically Example" on page 98](#)

Static Content Block

A static content block displays one identified content block on a Web page. The following table explains how to complete the server control properties to insert a static content block on a Web page.

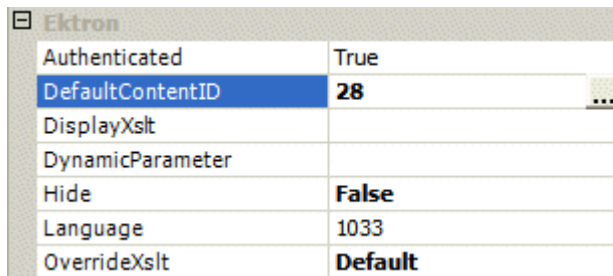
ContentBlock Server Control Properties

The following table explains the properties of the ContentBlock server control.

Note	The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.
------	--

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See <i>Also</i> : "Caching with Server Controls" on page 36 .	Integer
DefaultContentID	The ID of a content block that appears where you insert this server control. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DisplayXslt	Ignore for a non-XML content block.	String
DynamicParameter	Select None - Use Default . This parameter is used for dynamic content blocks.	String
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
OverrideXslt	Ignore for a non-XML content block.	must be one of these values: default, one, two, three

Property	Value	Data Type
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

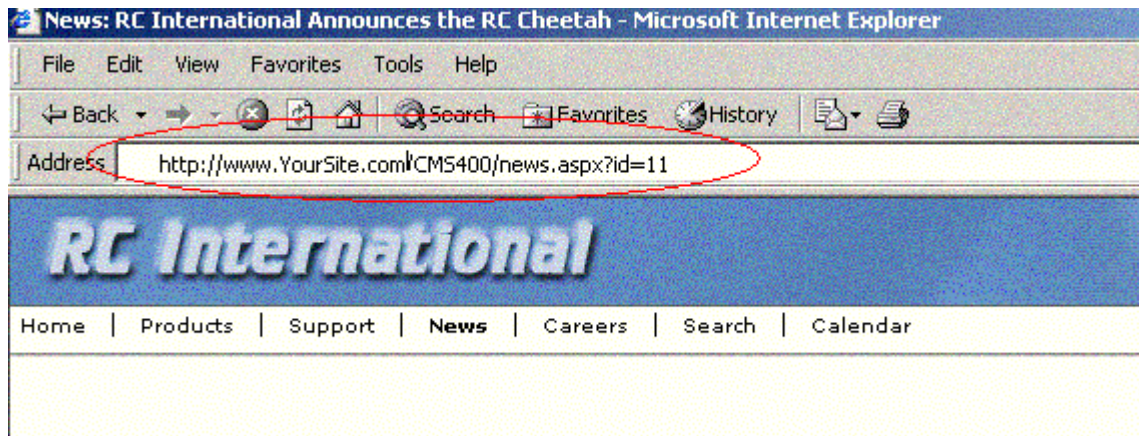


Ektron	
Authenticated	True
DefaultContentID	28
DisplayXslt	
DynamicParameter	
Hide	False
Language	1033
OverrideXslt	Default

In the above example, the server control retrieves content block id=28 and displays it in the browser.

Dynamic Content Block

Use a dynamic content block to display a content block whose ID is passed through a URL parameter. You would use this server control with a dynamic template.



The following table explains how to complete the ContentBlock server control's properties to insert a dynamic content block on a template.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
DefaultContentID	The ID of a content block that appears where you inserted this server control if no other content block is identified, or is not available. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DisplayXslt	Ignore for a non-XML content block.	String
DynamicParameter	Select id . When you do, this server control uses the content block passed as a URL parameter.	String
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
OverrideXslt	Ignore for a non-XML content block.	must be one of these values: default, one, two, three
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

The following example shows how the ContentBlock server control can be used to pass the id as a URL parameter.

Ektron	
Authenticated	False
DefaultContentID	1
DisplayXslt	
DynamicParameter	id
Hide	False
Language	1033
OverrideXslt	Default

When a user clicks a link that passes the content block ID as a URL parameter, that content block appears. If that content block is not available, content block 1 appears.

XML Content Block

Use an XML content block to display an XML content block on an Ektron CMS400.NET Web page.

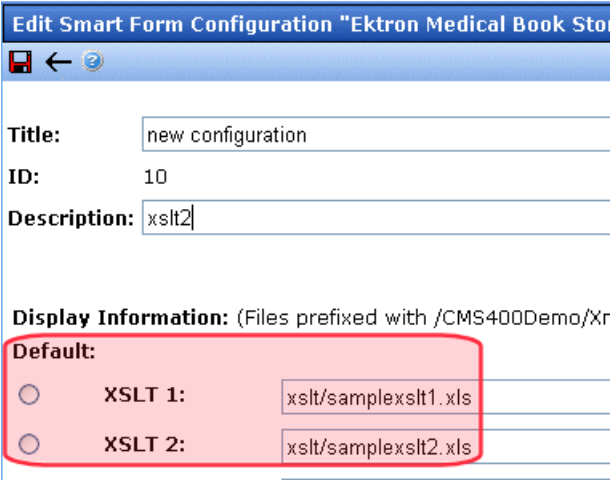
Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String
DefaultContentID	The ID of a content block that appears where you inserted this server control if no other content block is identified, or is not available. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	Integer
DisplayXslt	Specify an external XSLT file. See <i>Also: "The OverrideXslt and DisplayXslt Properties" on page 96</i>	String

Property	Value	Data Type
DynamicParameter	To make this content block dynamic, select id . When you do, this server control uses the content block passed as a URL parameter.	String
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
OverrideXslt	An internal Xslt file used to display XML content if one is not identified in the Xslt property. <i>See Also: "The OverrideXslt and DisplayXslt Properties" on page 96</i>	must be one of these values: default, one, two, three
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

The OverrideXslt and DisplayXslt Properties

The DisplayXslt property is optional. If used, it specifies an external XSLT file. If the DisplayXslt property is not defined, the OverrideXslt property specifies an XSLT identified in the Edit Smart Form Configuration screen. The following table provides more information about these properties.

To use this display XSLT	Enter this value for XSLT	Enter this value for Override XSLT
<p>The default XSLT specified in the Edit Smart Form Configuration screen (illustrated below). Note that, in this example, the XSLT Packaged option is the default XSLT, since it is selected.</p> <p>XSLT Packaged is the XSLT from the Edit Smart Form Configuration screen (that is, the XSLT created in the Data Designer).</p> 		Default
XSLT 1 from the Edit Smart Form Configuration screen		1
XSLT 2 from the Edit Smart Form Configuration screen		2
XSLT 3 from the Edit Smart Form Configuration screen		3

To use this display XSLT	Enter this value for XSLT	Enter this value for Override XSLT
An absolute or relative path to an XSLT file.	An external XSLT file not specified in the Edit Smart Form Configuration screen. For example <code>sample.xslt</code>	does not matter - if XSLT value exists, OverrideXSLT is ignored

See Also: The “Managing XML” chapter of the **Ektron CMS400.NET** Administrator manual

Retrieving the XML Structure of an XML Content Block

Retrieving the XML structure of XML content allows for greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a Content Block server control onto it.
3. Set the DefaultContentID to an XML content block.

WARNING! This does not work with HTML content blocks, as there is no XML structure to output.

4. Drag and drop a Text box on the Web form.
5. Set the TextMode property to MultiLine.

NOTE It is also recommended that you set the width of the text box to at least 400px.

6. On the code behind page, add the following line.
`TextBox1.Text = ContentBlock1.EkItem.Html`
7. Build the project.
8. View the form in a browser.
9. The XML structure of the content block appears in the text box.

Using the ContentBlock Server Control Programmatically Example

The following code will display a content block:

NOTE Before adding these lines of code, you need to drag and drop a literal box on your Web form.

```
Dim MyContentBlock As New ContentBlock
MyContentBlock.DefaultContentID = 8
MyContentBlock.Page = Page
MyContentBlock.Fill()
Literal1.Text = MyContentBlock.EkItem.Html
```

To display a content block with the content block title, do the following:

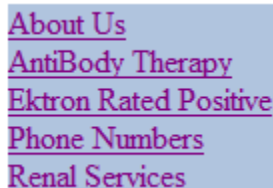
NOTE Before adding these lines of code, you need to drag and drop a two literal boxes on your Web form.

```
Dim MyContentBlock As New ContentBlock
MyContentBlock.DefaultContentID = 8
MyContentBlock.Page = Page
MyContentBlock.Fill()
Literal1.Text = MyContentBlock.EkItem.Title
Literal2.Text = MyContentBlock.EkItem.Html
```

ContentList Server Control

The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder.

When added to a template and visited, a ContentList Summary looks like this.



[About Us](#)
[AntiBody Therapy](#)
[Ektron Rated Positive](#)
[Phone Numbers](#)
[Renal Services](#)

Depending on the setting you choose for DisplayXslt, you can change the amount of information displayed for each content block. See Also: ["DisplayXslt" on page 102](#).

There are two options for the ContentList server control to display a content list. First, by defining a content list in the metadata of a content block and using that content block's ID in the **DefaultContentID** property. See Also: ["Using the DefaultContentID and MetaTag Properties" on page 105](#).

Second, by adding a comma delimited list of content blocks to **ContentIds** property. See Also: ["Using the ContentID Property to Display a Content List" on page 109](#).

Using the DefaultContentID and MetaTag process, an administrator typically sets up the Workarea portion of the process. Then a developer adds the server control to a Web form, assigns the ID of the content block and the metadata name to the appropriate properties. Because the content list in this process is assigned to a content's metadata, you can pass the content ID dynamically in a Web form and display a list for each content block you have defined.

Using the ContentIds process, a developer adds ContentList server control to Web form. Then, defines a list of content IDs in the ContentIds property. Once the server control is deleted or the list in the ContentIds property is deleted, It is not available and will have to be created again.

This subsection contains the following topics:

- ["ContentList Server Control Properties" on page 100](#)
- ["Using the DefaultContentID and MetaTag Properties" on page 105](#)
- ["Using the ContentID Property to Display a Content List" on page 109](#)

ContentList Server Control Properties

The ContentList server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Description	Data Type	Values
Authenticated	Indicates if you are logged in to the CMS Explorer. See <i>Also:</i> "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String	
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See <i>Also:</i> "Caching with Server Controls" on page 36	Integer	
ContentIds	A comma delimited list of content blocks IDs. See <i>Also:</i> "Using the ContentID Property to Display a Content List" on page 109	String	
DefaultContentID	Set content id value. Once set, content IDs are generated from the <code>MetaTag</code> value for this content. See <i>Also:</i> "MetaTag" on page 104 "Using the DefaultContentID and MetaTag Properties" on page 105	Integer	

Property	Description	Data Type	Values
Direction	Determines how to order the hyperlinks on the list. The sort field is determined by the OrderKey property.	either ascending or descending	ascending - they are arranged A, B, C or 1,2,3. descending - they are arranged. Z,Y,X or 3,2,1 If sorting by date, descending puts the most recent first.
DisplayXslt	Determines how information on the page is displayed.	String	None -databind only ecmNavigation - lists the title of each content block See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists the title of each content block plus the content summary See Also: "Example of ecmTeaser Display" on page 85 ecmUnOrderedList - sorts the list in no particular order. Shows the title and content summary Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the list summary. For example, to display content inside a web server control such as a GridView.	Boolean	True = Get and display HTML for each content block in the list summary False = Do not get and display HTML.
Hide	Used to hide output of the list summary in design time and run time.	Boolean	True = Hide results False = show results
Language	Set a language for viewing the list summary. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer	

Property	Description	Data Type	Values
LinkTarget	Defines the way a link acts when a link is clicked.	String	<p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>

Property	Description	Data Type	Values
MetaTag	<p>Specify a Metadata definition whose type is Content Selector. When you do, the associated list of content items will appear where you place the server control.</p> <hr/> <p>Warning!: You cannot insert other metadata types.</p> <hr/> <p>This works with the DefaultContentID property. See Also: "DefaultContentID" on page 101 "Using the DefaultContentID and MetaTag Properties" on page 105</p> <p>For more information about using metadata to assign a list of related content to a content item, see the Ektron CMS400.NET Administrator manual section "Managing Content" > "Metadata" > "Types of Metadata" > "Related Content Metadata."</p>	Integer	
OrderKey	Sort the list by one of the values.	must be one of the values	<p>Title - the order of the content's title.</p> <p>DateModified - the last date the content was modified.</p> <p>DateCreated - the date the content was created.</p> <p>LastEditorFname - the last editor's first name.</p> <p>LastEditorLname - the last editor's last name.</p> <p>OrderOfTheIds - preserves the content ID order based on the list in the ContentIds property.</p>

Property	Description	Data Type	Values
Random	Set to True if you want to randomly display one content block in the specified folder. The content changes each time a user views the page.	Boolean	True False
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean	True False
WrapTag	Allows a developer to specify a server control's tag. The default is Span .	String	Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.

Using the DefaultContentID and MetaTag Properties

You define a content list in the metadata of a content block. Then, using the ContentList server control, you can display the list of content blocks associated with the content. The ContentList server control pulls information defined in the metadata of the content to display the list of content.

The steps below show an example of using the DefaultContentID and MetaTag properties to create a content list.

1. In the CMS400.NET Workarea, create a metadata definition, in the **Name** field, enter a title of the data. This name is used in the MetaTag property for the server control.
2. Fill in all of the standard fields. See the **Ektron CMS400.NET** Administrator Manual section "Managing Content" > "Metadata" > "Working with Metadata Definitions" > "Adding a Metadata Definition" > "Metadata Definition Fields".
The **Type** must be **Content Selector**.

Add Metadata definition

Name: TestContentList

Type: Searchable Property

Editable: Searchable Property

Display in Ektron Explorer: Searchable Property, Meta Tag, HTML Tag, Collection Selector, **Content Selector**, File Selector, Hyperlink Selector, Image Selector, ListSummary Selector

3. Add the definition to all appropriate folders. See the **Ektron CMS400.NET** Administrator Manual section "Managing Content" > "Metadata" > "Working with Metadata Definitions" > "Assigning Metadata to a Folder"
4. For each content item to which you want to associate related content, access its **Metadata** tab and identify the related item(s).
For example, a list of content blocks associated with motorcycle helmets. You want this list to appear whenever a helmet is being viewed.
To set this up, you would
 - edit the content block
 - click its **Metadata** tab
 - find the metadata definition for the content list
 - click **Change**

Content **Summary** **Metadata** **Schedule** **Comment** **Web Alerts** **Templates**

TestContentList: (Content)

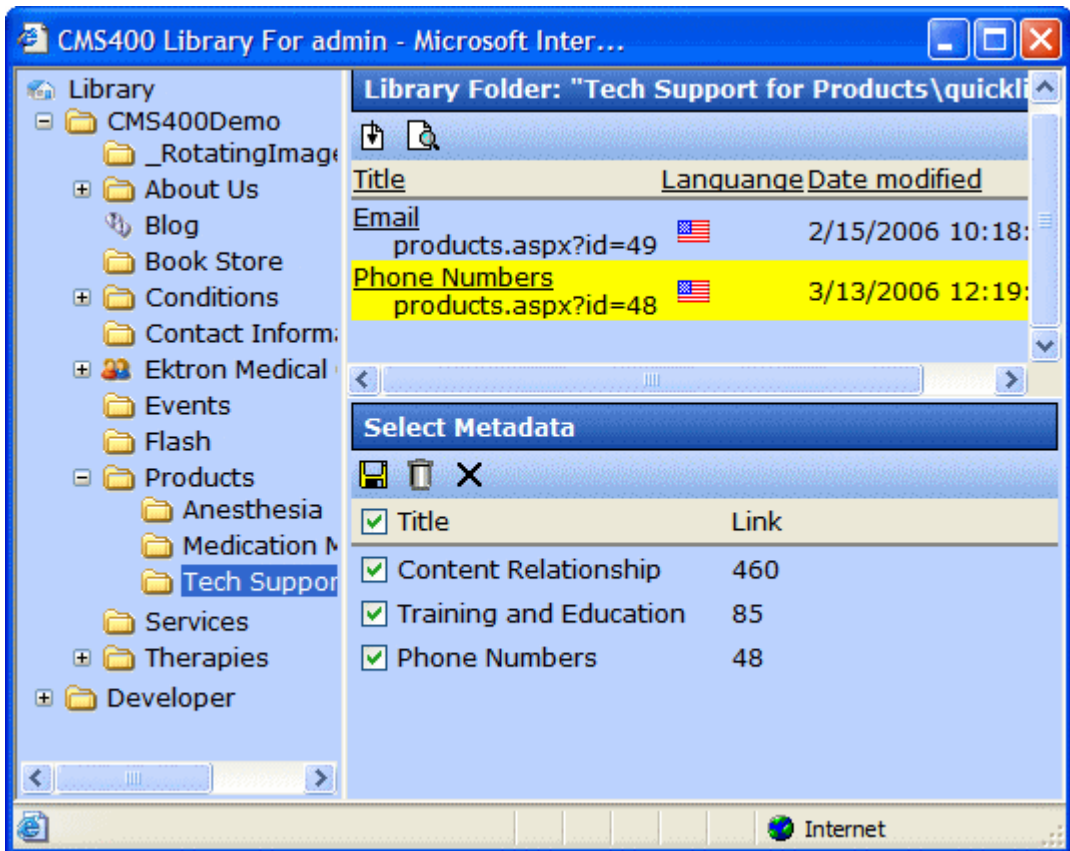
None selected

Change Clear

Search Data

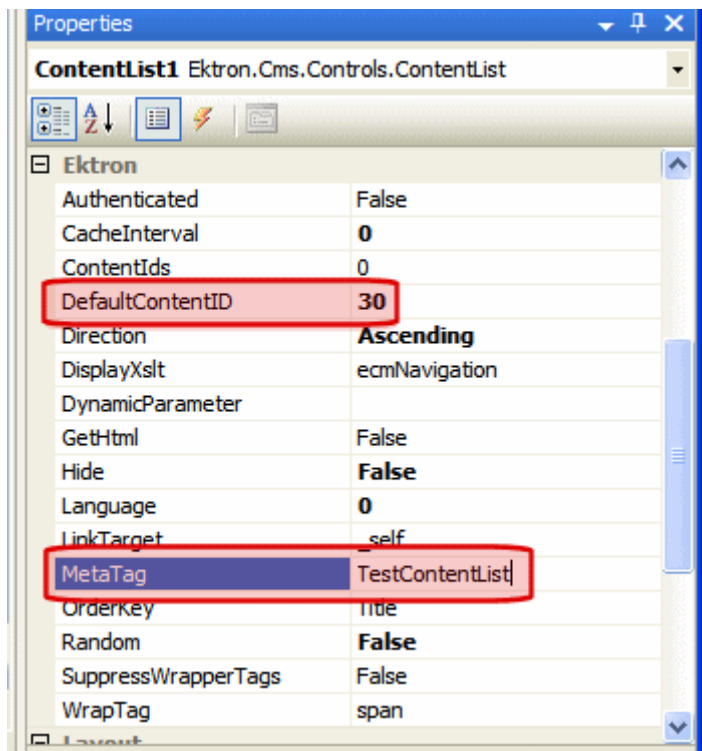
wellness: (No Selection) ▼

- select the content blocks to be in the list

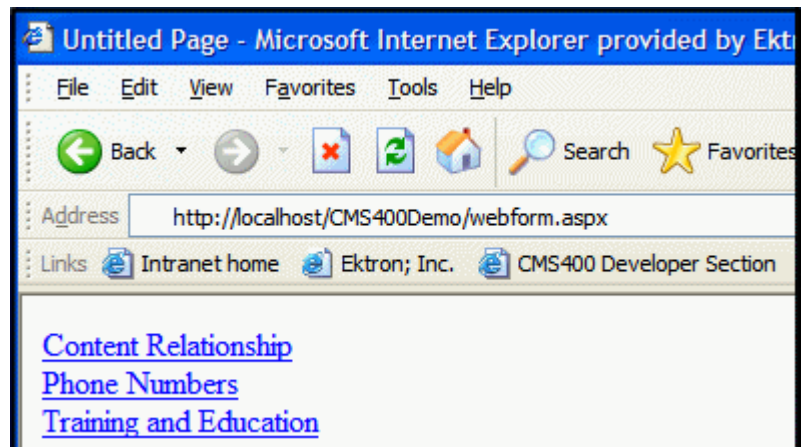


- On a Web form in Visual Studio, add the ContentList server control.
- Set the **DefaultContentID** property to the content block's ID.

7. Add the name of the Metadata field in the Workarea to the **MetaTag** property.

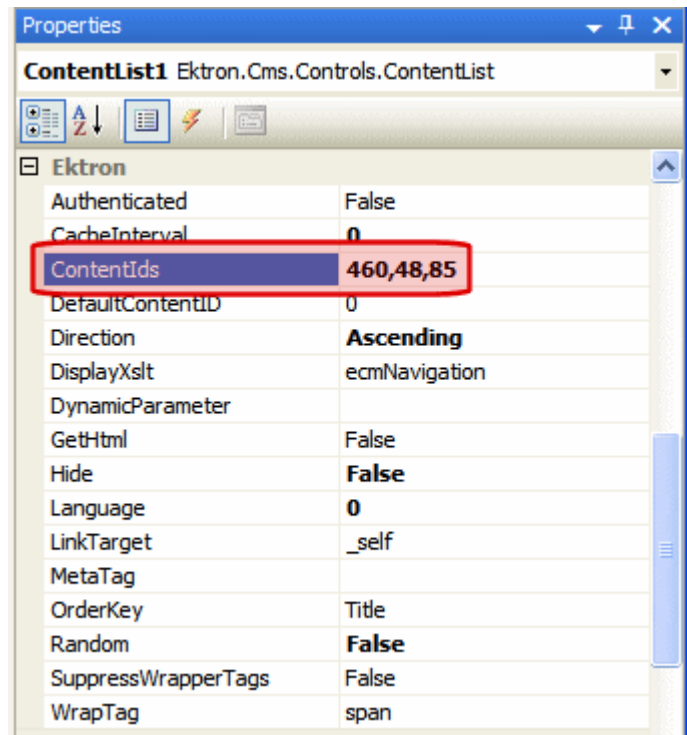


8. Set all other properties using the table in "ContentList Server Control Properties" on page 100.
9. Once the Web form is saved, navigate to the page and the content list is displayed.

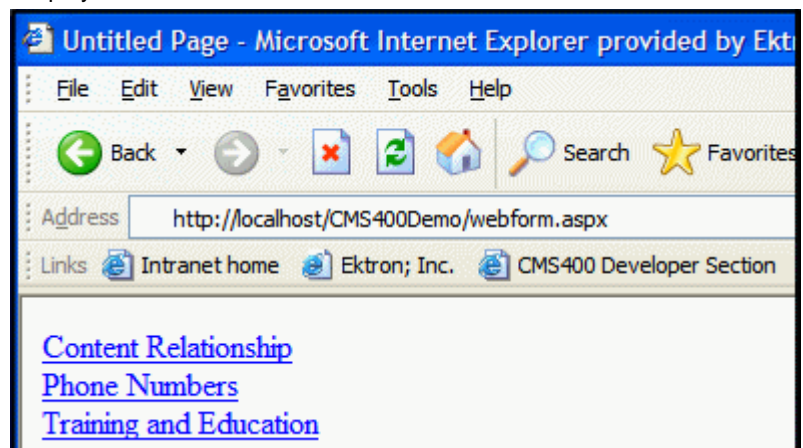


Using the ContentID Property to Display a Content List

1. On a Web form in Visual Studio, add the ContentList server control.
2. Add a comma delimited list of content IDs to the **ContentIds** property.



3. Set all other properties using the table in "ContentList Server Control Properties" on page 100.
4. Once the Web form is saved, navigate to the page and the content list is displayed.



ContentRating Server Control

Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment.

The control lets a site visitor rate a content item only once. On subsequent visits to the page, a message informs the visitor that he has previously rated the page, which can optionally display a summary of ratings for it. As a result, the control has two appearance modes: *initial* and *complete*.

Ektron CMS400.NET maintains and reports content ratings data. For more information, see the **Ektron CMS400.NET** Administrator Manual section "Managing Content" > "Content Rating."

Content Rating Server Control Properties

The Ektron-specific Content Rating server control properties are divided into two groups:

- *Ektron* - settings that affect properties, such as the display
- *Ektron labels* - settings that affect the text labels within the control

These properties are described below.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Ektron Properties

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Boolean

Property	Value	Data Type
CompleteGraph Display	Specify a value to indicate how the ratings graph appears if a site visitor has already rated content. <ul style="list-style-type: none"> • horizontal - graph appears to the right of the AlreadyRatedMessage • vertical - graph appears below the AlreadyRated-Message • none - graph does not appear 	String
DefaultContentID	The ID of a content block being rated by this server control. It typically would be the content directly above the rating scale if no other content block is identified, or is not available. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer
DynamicParameter	To make this content rating control dynamic, select id . When you do, this server control is attached to the content block passed as a URL parameter.	String
GraphBarColor	The color of the bars in the graph.	String
Hide	Select False to display this server control on the page. Select True to suppress it.	Boolean
InitialGraphDisplay	Specify a value to indicate how the ratings graph appears if a site visitor has not yet rated content. <ul style="list-style-type: none"> • horizontal - graph appears to the right of the content rating scale • vertical - graph appears below the content rating scale • none - graph does not appear 	String
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer

Property	Value	Data Type
Padding	Enter the number of pixels used to create space between the rating scale and the graph. <u>Note: If you do not display the graph, this setting has no effect.</u>	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element. Div - The <code><div></code> tag is used when you want to apply attributes to a block of code. Custom - Lets you use custom tag.	String

Ektron Label Properties

The label properties determine the text that appears in the server control. Most of them are illustrated below.

Note that the first illustration and table describe the initial graph, while the second illustration and table describe the complete graph.

Initial Graph Display

Feedback Header

How would you rate the usefulness of this content?

Bad Label

Poor 1 2 3 4 5 6 7 8 9 Outstanding

Good Label

Optional: Tell us why you rated the content this way

Characters Remaining Label

Characters Remaining: 1024

Submit

Please select a rating before submitting your feedback.

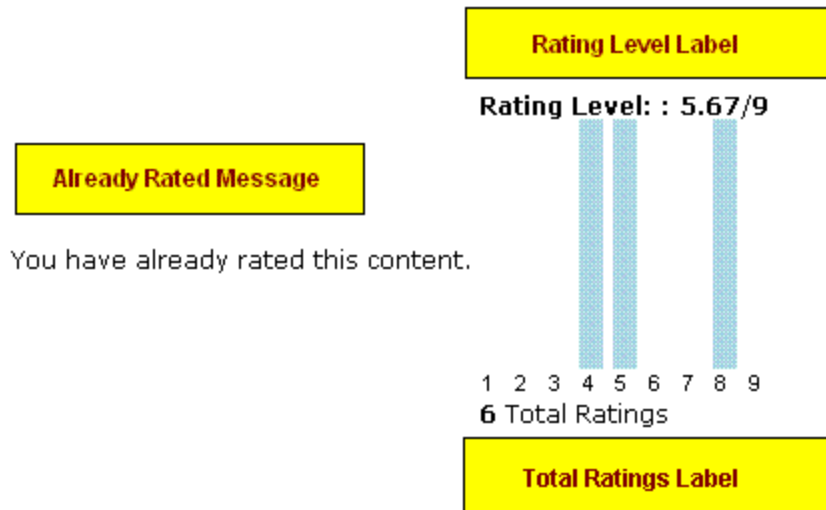
User Comments Header

Validation Message

Initial Graph Properties

Property	Description	Default Text
BadLabel	The low end of the rating scale	Poor
CharsRemainingLabel	Number of additional characters user may enter into comment	Characters Remaining:
CookiesRequired Message	Appears if browser cookies have been disabled	We're sorry, but you must enable cookies in your browser to rate this content.
FeedbackHeader	Introduces rating scale	How would you rate the usefulness of this content?
GoodLabel	The high end of the rating scale	Outstanding
RatingLevelLabel	Appears at the top of the graph	Rating Level:
ThankYouMessage	Appears after site visitor selects a number, optionally enters a comment, then presses Submit .	Thank you for your feedback.
TotalRatingsLabel	Appears below the graph, and lists the number of rating submissions received.	Total Ratings
UserCommentsHeader	Introduces free-text comment area below rating scale	Tell us why you rated the content this way
ValidationMessage	Appears on initial display if user clicks Submit button before selecting rating number	Please select a rating before submitting your feedback.

Complete Graph Display



Complete Graph Properties

Property	Description	Default Text
AlreadyRatedMessage	Appears in place of the rating scale if the user previously rated the page	You have already rated this content.
RatingLevelLabel	Appears at the top of the graph	Rating Level:
TotalRatingsLabel	Appears below the graph, and lists the number of rating submissions received.	Total Ratings

DesignTimeDiagnostic Server Control

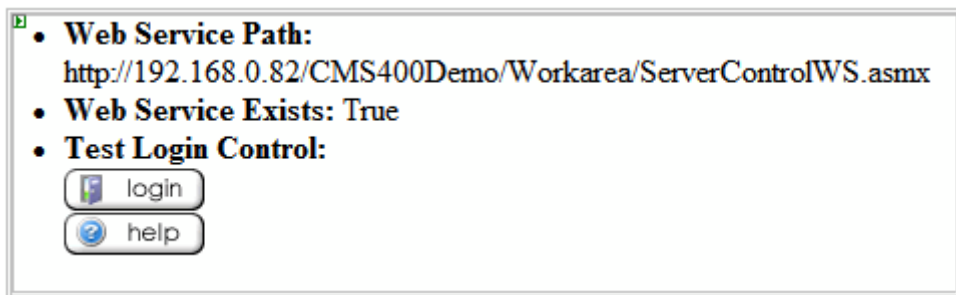
IMPORTANT!

This control is for use in design-time only, nothing is rendered at run-time.

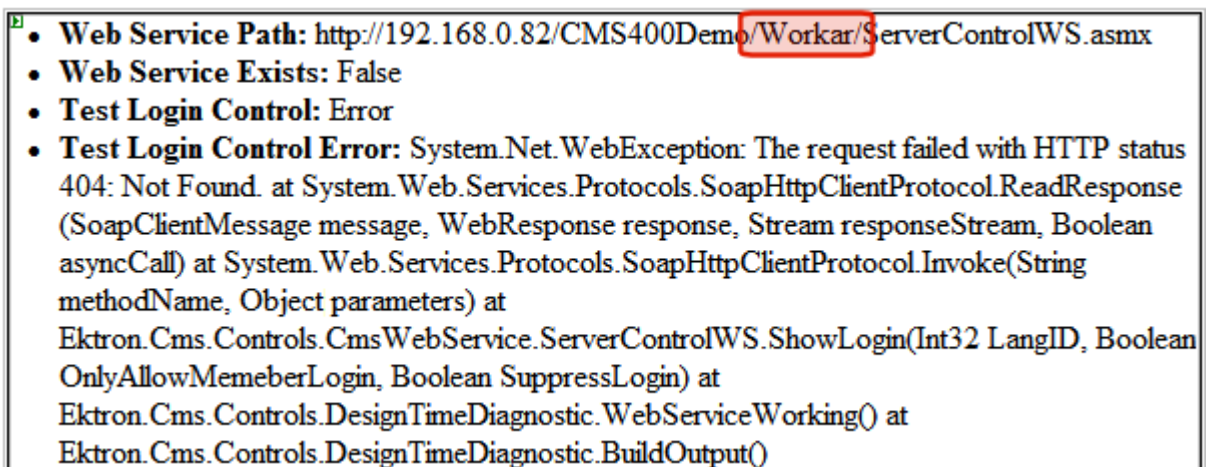
The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information:

- **Web Service Path** - displays the server control Web service path in your web.config file
- **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False.
- **Test Login Control** - displays the Login server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.

Below is an example of the control on a page connected to the Web service.



Below is an example of the control on a page not connecting to the Web service. Note that the path is not the correct path. It should be: `http://192.168.0.82/CMS400Demo/Workarea/ServerControlWS.asmx`



DesignTimeDiagnostic Server Control Properties

The DesignTimeDiagnostic server control has no definable ektron properties.

FormBlock Server Control

The FormBlock server control displays a content block associated with a form.

IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the Ektron CMS400.NET Administrator manual "Managing Library Assets" > "Updating Default Template for Multiple Quicklinks."

When added to a template and visited, the form content block might look like this. You can change the display to suit your needs by modifying its properties.

Absence Request Form

Name:	<input type="text"/>
E-mail Address:	<input type="text"/>
Position	<input type="text"/>
Department:	<input type="text" value="Select Department"/>

Reason For Absence?

☐ Floating Holiday

☐ Vacation

☐ Sick Leave

☐ FMLA

☐ Other:

Dates of Absence: to

Total Days Absent:

☐ With Pay

☐ Without Pay

Comments:

This subsection contains the following topics:

- ["FormBlock Server Control Properties" on page 118](#)

- ["Automatic versus Manual Generation of Form Tags" on page 119](#)

FormBlock Server Control Properties

The following table explains the properties of the FormBlock server control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see [VisualStudio help](#).

Property	Value	Data Type
AddValidation	Set to True to add Validation to your main runat=server form.	Boolean
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.	Integer
DefaultFormID	The ID of a FormBlock that appears where you inserted this server control if no other form block is identified. If you don't know the ID number of the form block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.	Integer
DynamicParameter	To make this form block dynamic, select id . When you do, this server control uses the form block passed as a URL parameter.	String
FormName	Set this value to the name of your main runat=server form.	String
Hide	Used to hide a form block in design time and run time. True = Hide form block False = Show form block	Boolean

Property	Value	Data Type
Include Tags	Determines if tags are generated automatically of manually. See Also: "Automatic versus Manual Generation of Form Tags" on page 119	Boolean
Language	Set a language for viewing form content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Automatic versus Manual Generation of Form Tags

When using the FormBlock Server control's Include tags property, you have two options.

Option	Recommended if you are developing	How to Enable
Let Ektron CMS400.NET generate form tags automatically	A pure script .aspx page	Set Include Tags property to True .
Modify HTML form tags in the .aspx file	An .aspx page and associated code-behind web form	Set Include Tags property to False . Here is the default .NET generated form tag: <pre><form id="Form1" method="post" runat="server"></pre> Modify the form tag as indicated in red: <pre><form id="Form1" method="post" runat="server" OnSubmit="EkFmValidate(this); "></pre>

Forum Server Control

The Discussion Board feature provides a forum where topics can be discussed on your Web site. An administrator creates a topic and posts a starter question. Any site visitor can then reply.

Use the Forum server control to place a Discussion Board on a Web page.

For more information about the Forum server control, see the **Ektron CMS400.NET** Administrator manual chapter "Discussion Boards." Within that chapter, be sure to follow the guidelines in the section "Inserting the Forum Server Control."

WARNING!


After placing a Forum server control on a page, follow the procedure described in "Updating the Page Command" on page 121. Otherwise, the user may get an error when posting a reply.

Server Control Properties

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Ektron Properties

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	Boolean
BoardPath	If you know the path to the discussion board relative to your site root, enter it here. For example, \CMS400Demo\MedicalCenterForum. If you don't know the path, click the Ellipses button (), then sign in, browse to and select the discussion board. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String

Property	Value	Data Type
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36	Integer
Hide	Select False to display this server control on the page. Select True to suppress it.	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element. Div - The <code><div></code> tag is used when you want to apply attributes to a block of code. Custom - Lets you use custom tag.	String

Updating the Page Command

To prevent an error from appearing when a user posts a reply to the Web page that hosts the Forum server control, follow these steps.

1. Open the Web form onto which you inserted the Forum server control.
2. Access Source view.
3. Go to the top line of the Web form, which begins with `@ Page`.

```
<%@ PageLanguage="VB"AutoEventWireup="false" CodeFile="Default2.aspx.vb" Inherits="Default2" %>
```

4. Near the end of the line, enter `validaterequest=false`.
5. Build the page.

HTMLEditor Server Control

Introduction

This server control provides the ability to edit CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.

WARNING!

After placing an HTMLEditor server control on a page, follow the procedure described in "Updating the Page Command" on page 121. Otherwise, the user may get an error when submitting the content.

The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page.

- **[ControlName]_Action** = the action to take - save, check in, undo checkout
- **[ControlName]_ContentTitle** = the content title
- **[ControlName]_ContentId** = the ID of the content
- **[ControlName]_Language** = the language of the content

The content title caption is surrounded with span tags that have this ID:

- **[ControlName]_TitleLabel**

All information for all content information fields (except the content itself) is surrounded with a div tag with this ID:

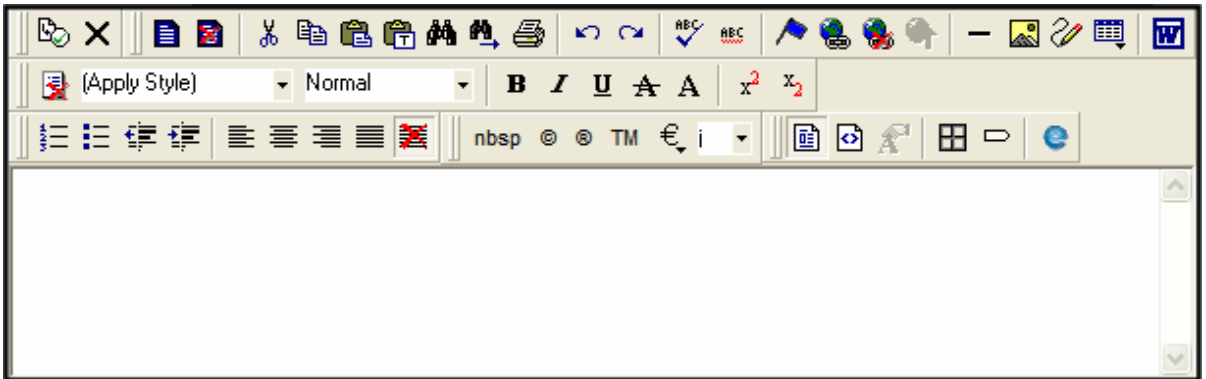
- **[ControlName]_ContentInfo**

The valid values for the [ControlName]_Action value are:

- **Publish** - submit the posted content for publication
- **CheckIn** - check in the content
- **Save** - save the content and redisplay in the editor
- **UndoCheckout** - undo the checkout of the content

The server control displays the Ektron CMS400 HTML editor on a Web page. By adding the HTMLEditor to a Web page, site visitors can create or edit HTML

content for a site or project. Below is an example of the editor with the default toolbars.



The HTML editor server has two modes.

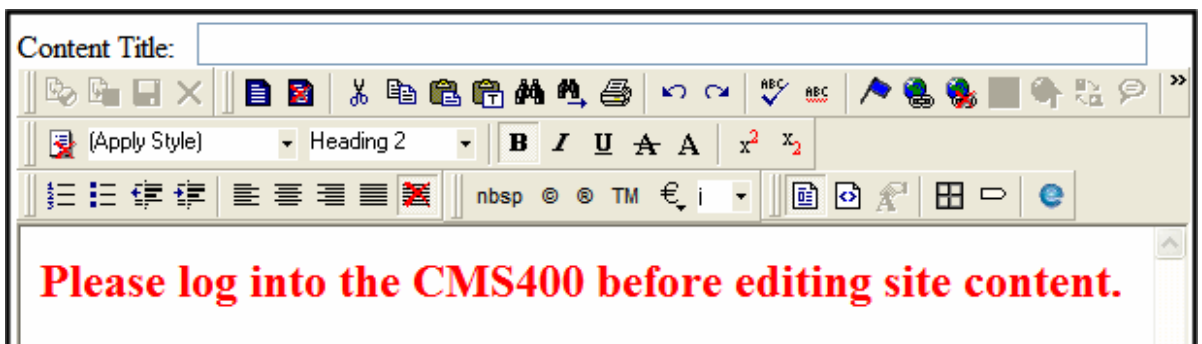
- working with Ektron CMS400.NET - See ["Working with Ektron CMS400.NET Content" on page 123](#)
- working as an independent Html editor - See ["Working with Non Ektron CMS400.NET Content" on page 124](#)

Working with Ektron CMS400.NET Content

When you work with Ektron CMS400.NET, site visitors need to be logged in. When logged in, a site visitor can edit existing content or create new content.

To edit existing content, the content's numeric ID must be passed to the ContentId property. If you enter an ID in the ContentId property, the HTML editor assumes you are working with CMS400.NET. To create new content, leave ContentId blank.

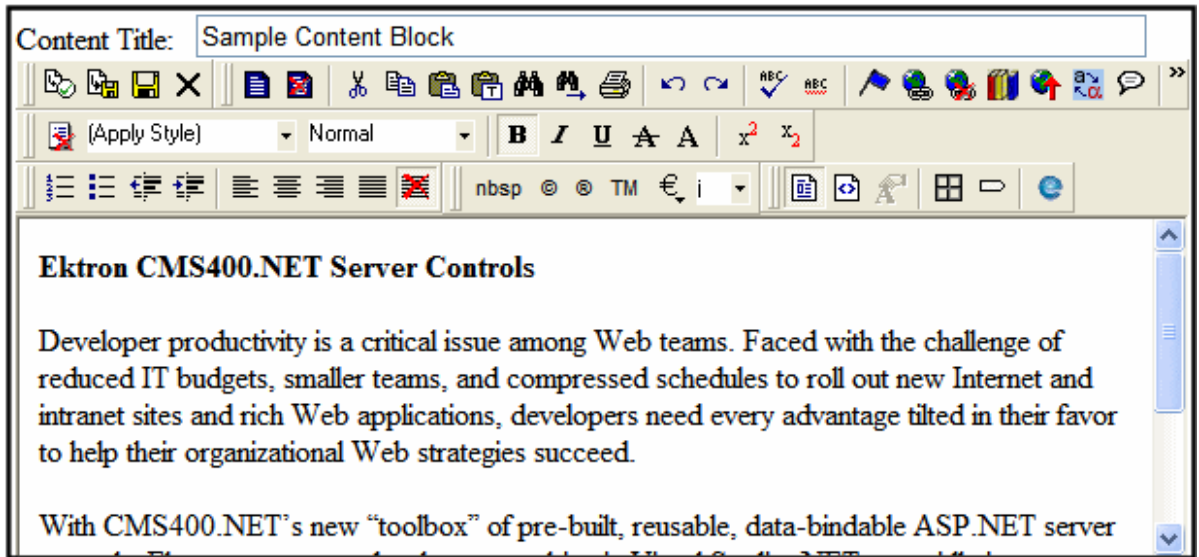
When content is loaded into the editor but the site visitor is not logged in, the editor displays the following message:



Notice the CMS toolbar's Publish, Check-in, Save, and Cancel buttons are grayed out.



Once a site visitor logs in, the content loads and the CMS toolbar is active.



Working with Non Ektron CMS400.NET Content

If a user is not logged to Ektron CMS400.NET, the HTML editor is used as an independent editor. For examples of integrating the Html Editor with your Web site, see the following files in `\webroot\CMS400Demo\Developer\ContentBlock`.

- **EditAnyHtmlSample.aspx.vb** - shows an example when not logged in
- **EditHtmlSample.aspx.vb** - shows an example when logged in

Uploading Files with an Action Page

When working with non-CMS400.NET content, create an action page to handle the uploading of files. In addition, edit the `cms_config.aspx` file to show the location of the new action page. To edit `cms_config.aspx`, follow these steps:

1. Open `cms_config.aspx`, located in `site root\Workarea\ewebeditpro\`.
2. Between the `<Mediafiles>` tags, find `<transport allowupload="true" type="ftp" xfer="binary" pasv="true">`
3. Change the `Type` element's value from `ftp` to the location of your action page. For example: `Type="CMS400Demo/myfileselect.aspx"`
4. Save and close the file.

HTML Editor Server Control Properties

The following table explains the properties of the HTML Editor server control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Archived	Signifies that the selected content is archived. For more information on archived content, see the User Manual section "Scheduling Content to Begin and End" > "Setting Archive Options".	Boolean
Authenticated	A read-only property that indicates if you are logged in to the CMS Explorer and can use it to browse to content, collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.	String
CmsSitePath	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.	String
Content ID	Specifies the content ID to load. This value must match a value that exists in the CMS database. The selection of content is similar to that in the ContentBlock control. The content can only be accessed if the user is logged in. This value can be set only on editor load, before the editor and contents are rendered. After editor load, this value is ignored. If this value is 0, then the code behind must handle making the calls that check out content and setting the content at edit time by placing it into the .Text property for the editor. A hidden field, named EktContentID_[ControlName], is created to hold this value. It is posted up with the rest of the content.	Integer
ContentType	This value describes the type of content the selected ID refers to. If this value is blank, then the content is a custom type, to be defined type, or unsupported type. These are the possible types: HTML Form	String

Property	Value	Data Type
EditorReadyCall	<p>This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content.</p> <p>This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:</p> <pre>function initEditorIsReady(sEditor, strURL, strAutoURL) { ... }</pre> <p>where the "initEditorIsReady" string is given to this property. If this string is empty, the default CMS method is called.</p>	String
EnableUpload	<p>If this is 'true' then the user can upload locally selected images and linked files that they have put into their content. If enabled, the automatic upload mechanism will allow upload the local files automatically with minimal user intervention. If this value is 'false' then the site developer must provide an upload page, such as the CMS library, where the upload can occur.</p>	Boolean
IncludeContentInfo	<p>If true, this will include the information about the content in fields posted with a form post. (This fields are defined in the Introduction of this section.)</p> <p>Set the value to 'false' if the page will maintain these values or the page is just using the editor to edit general formatted text.</p> <p>If the information must be posted, but not shown to the end user, then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle property to false.</p>	Boolean
Language	<p>Specifies the ID of the language to use for the content to load. This value must match a value that exists in the CMS database.</p>	Integer
ShowCMSToolbar	<p>Controls whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing the CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation. <i>See Also:</i> "ShowCMSToolbar" on page 127</p>	Boolean

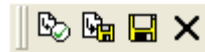
Property	Value	Data Type
StyleSheet	Assigns a single sheet to the edited content.	String
Test	The HTML text being edited.	String
ToolbarLevel	This specifies the level of the toolbar feature set offered to the user. This defines the amount of functionality offered to the user. See Also: "ToolbarLevel" on page 128 .	String
ToolbarResetCall	<p>This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars.</p> <p>This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:</p> <pre>function initToolbarCommands(sEditor, strURL, strAutoURL) { ... }</pre> <p>where the "initToolbarCommands" string is given to this property.</p> <p>A toolbar reset is only called when the toolbar actually resets. This often happens when the editor starts, however, if the user has been given the opportunity to save their own toolbar setup then this will not be called on startup. It is always called when the user selects to reset the toolbar.</p> <p>If this string is empty, the default CMS method is called.</p>	String
TranslationFile	The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select from the series of default translation files.	String

ShowCMSToolbar

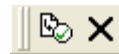
NOTE For information on the Toolbar buttons and there functions, see the User Manual Section > "Toolbars".

The CMS Toolbar appears when the ShowCMSToolbar property is set to true. There are two versions of the toolbar that appear:

- site visitor is logged into Ektron CMS400.NET



- site visitor is not logged in



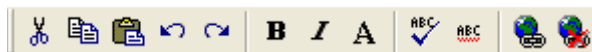
ToolbarLevel

NOTE

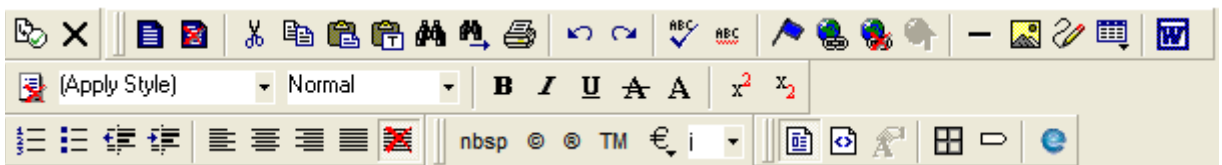
For information on the Toolbar buttons and there functions, see the User Manual Section > "Toolbars".

This property specifies the level of the toolbar feature set offered to site visitors. It may be that they need a minimal set of functionality to keep the editing simple. Or, they need full functionality for a major edit. The choices available are:

- **None** - No toolbars are shown.
- **Reduced** - Allows for minimal changes to the text.



- **Default** - Loads the toolbars typically need for working with content.



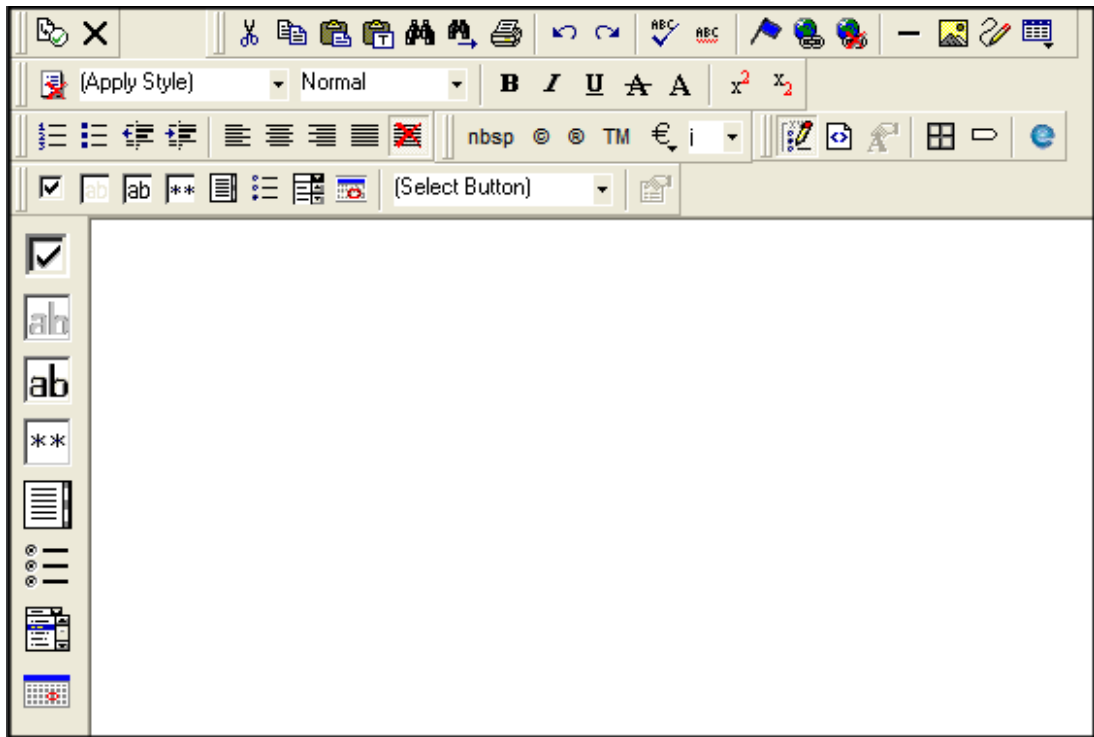
- **Calendar** - Loads the Paragraph and Paragraph Format toolbars.



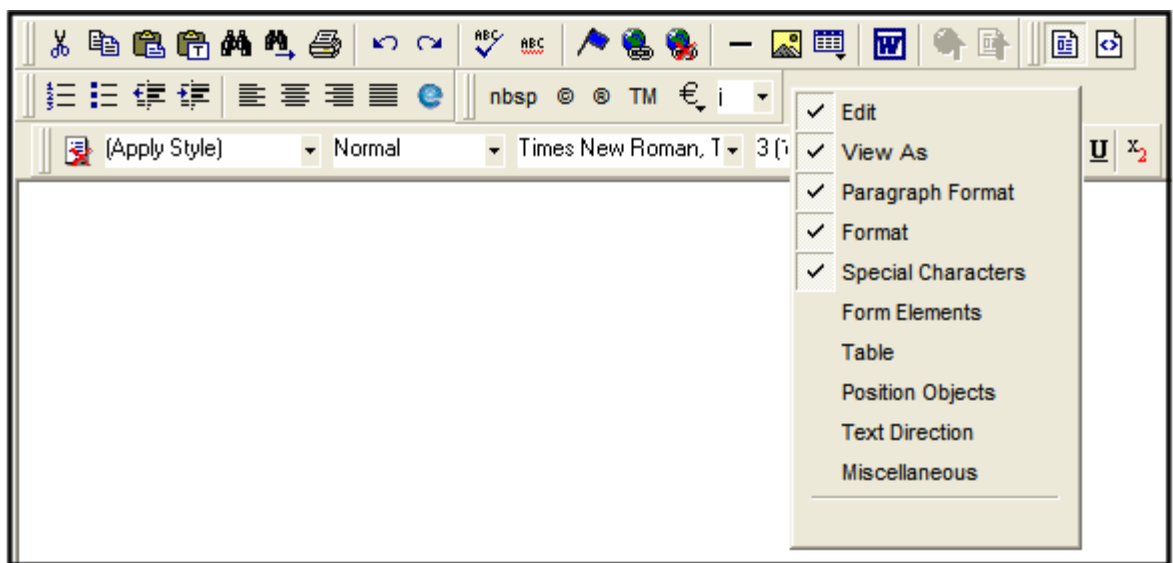
- **FormDesign** - Loads the toolbars that are needed to work with forms.

IMPORTANT!

The Data Designer is not supported when using the HTML editor server control.



- **Custom** - Allows a site user to define the toolbars that appear by right clicking on the toolbar area of the screen.



LanguageAPI Server Control

The LanguageAPI server control lets a developer force a particular language for a Web site. You can do this simply by dropping the server control on the page and then choosing a language in the SiteLanguage property box. You can also override site language logic by programmatically using the LanguageAPI server control to detect the browser's language, and display the site in that language.

This subsection contains the following topics:

- ["LanguageSelect Server Control vs. LanguageAPI Server Control" on page 130](#)
- ["LanguageAPI Server Control Properties" on page 130](#)
- ["Code Behind Only Properties" on page 131](#)

LanguageSelect Server Control vs. LanguageAPI Server Control

The LanguageSelect server control lets users view your site in the language they choose. The LanguageAPI server control, on the other hand, forces users to view a site in a particular language. For information on LanguageSelect, see ["LanguageSelect Server Control" on page 133](#).

LanguageAPI Server Control Properties

The LanguageAPI server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
SiteLanguage	Sets the site language. Runs through the IsValid function to verify that the language is an active language in the system.	String
SiteLanguageID	Numeric value of the site language. This is the property you will use if you are using code behind to set the sites language ID. Example: 1036 = French For a list of supported languages, see The Administrator Manual > "Appendix A: List of Supported Languages"	Integer

Code Behind Only Properties

The following is a list of LanguageAPI specific properties that can only be used programmatically.

Property	Value	Data Type
DefaultLanguage	Read only. What the default language of the site is set to. For example, the demo site is "English (Standard)".	String
DefaultLanguageID	Read only. Returns the value that is the default language id of the site. For example, the demo site is "1033" for English.	Integer
GetLanguage	This method returns a string. Pass in a valid language ID and it will return the language name.	Integer Argument
GetLanguageID	This method returns an integer. Pass in a valid language name and it will return the language ID.	String Argument
IsValid	This method returns a boolean. You can pass in a language ID or a string and it will tell you if the system is supporting it.	Argument
LanguageIDList	Read only. Lists all the language IDs that are activated in Ektron CMS400.NET . For information on how to enable languages, see The Administrator Manual > "Multi-Language Support" > "Working with Multi-Language Content" > "Determining Which Languages are Available".	Array of Integers
LanguageTitleList	Read only. Lists all the languages that are activated in Ektron CMS400.NET .	Array of Strings
MultiLanguageEnabled	Read only. Tells if the site supports multi-language mode. True = Multi-language enabled False = Multi-language not enabled Multi-language is enable by default in Ektron CMS400.NET . To disable Multi-language, see Administrator Manual > "Multi-Language Support" > "Working with Multi-Language Content" > "Disabling Support for Multiple Language Content".	Boolean

Using the LanguageAPI Server Control Programmatically

This example uses a logo that is not managed through Ektron CMS400.NET. It retrieves the current language from the LanguageAPI control, and uses that information to choose the logo version to display. The codebehind looks like this.

```
Select Case LanguageAPI1.CurrentLanguageID

    Case 1031

        Image1.ImageUrl = "germanlogo.gif"

    Case 1033

        Image1.ImageUrl = "englishlogo.gif"

    Case 1036

        Image1.ImageUrl = "frenchlogo.gif"

End Select
```

LanguageSelect Server Control

The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page.

This control lets a visitor to your Web site select a language in which to view the site. Here is what the control looks like when published on a Web page.



It lists all languages selected on the **Settings > Configuration > Language settings** screen. (For more information, see the “Multi-Language Support” chapter of the **Ektron CMS400.NET Administrator** manual.)

You can place this control in any location of any page on your site. On the sample site, it appears in the lower right corner of the home page.

This subsection contains the following topics:

- ["LanguageSelect Server Control vs. LanguageAPI Server Control" on page 133](#)
- ["LanguageSelect Server Control Properties" on page 133](#)

LanguageSelect Server Control vs. LanguageAPI Server Control

The LanguageSelect server control lets users view your site in the language they choose. The LanguageAPI server control, on the other hand, forces users to view a site in a particular language. For information on LanguageAPI, see ["LanguageAPI Server Control" on page 130](#).

LanguageSelect Server Control Properties

The following table explains the properties of the LanguageSelect control.

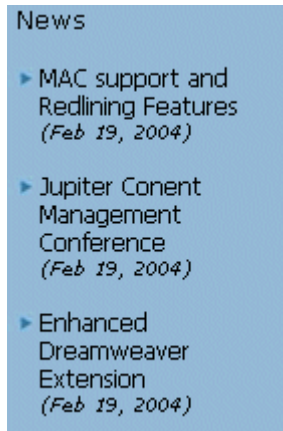
Note	The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.
------	--

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
LabelName	Lets you define the label next to the language select dropdown box. The default label is Language	String
Language	Set a language for the Language Select Box. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

ListSummary Server Control

The ListSummary server control is used to display a list of content blocks on a Web page.

When added to a template and visited, a List Summary looks like this.



It displays each content block's title and summary information. You can modify the display to suit your needs by modifying its properties.

This subsection contains the following topics:

- ["ListSummary Server Control Properties" on page 135](#)
- ["OrderKey Property" on page 140](#)
- ["List Summary Example" on page 141](#)
- ["Retrieving the XML Structure of a List Summary" on page 142](#)

ListSummary Server Control Properties

The ListSummary server control properties are described in this table.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Description	Data Type	Values
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderID property. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</i>	String	
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See <i>Also: "Caching with Server Controls" on page 36</i>	Integer	
ContentParameter	Checks the QueryString for this value and replaces the list summary with a content block when specified. Leave blank to always display the list summary.	String	

Property	Description	Data Type	Values
ContentType	Determines the type of content that appears in the list summary. The default is Content.	String	<p>AllTypes - displays all of the content types for the given folder.</p> <p>Content - displays a list of content blocks.</p> <p>Forms - forms appear in the list summary.</p> <p>Archive_Content - archived content blocks appear in the list summary.</p> <p>Archive_Forms - archived forms appear in the list summary.</p> <p>Assets - assets, such as offices documents, appear in the list summary.</p> <p>Archive_Assets - archived assets appear in the list summary.</p> <p>LibraryItem - library items appear in the list summary.</p> <p>NonLibraryContent - all types of content appear in the list summary except for library items.</p>
Direction	How to order the hyperlinks on the list. The sort field is determined by the OrderKey property.	either ascending or descending	<p>ascending - they are arranged A, B, C or 1,2,3.</p> <p>descending - they are arranged. Z,Y,X or 3,2,1</p> <p>If sorting by date, descending puts the most recent first.</p>
DisplayXslt	Determines how the information on the page is displayed	String	<p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder See Also: "Example of ecmNavigation Display" on page 85</p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary See Also: "Example of ecmTeaser Display" on page 85</p> <p>Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page</p>

Property	Description	Data Type	Values
FolderID	<p>The folder that contains the content blocks which appear in the list summary. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p> <p>The Recursive property determines whether content blocks in this folder's child folders also appear.</p>	ID	Integer
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the list summary. For example, to display content inside a web server control such as a GridView.	Boolean	<p>True = Get and display HTML for each content block in the list summary</p> <p>False = Do not get and display HTML.</p>
Hide	Used to hide output of the list summary in design time and run time.	Boolean	<p>True = Hide results</p> <p>False = show results</p>
Language	Set a language for viewing the list summary. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer	Integer

Property	Description	Data Type	Values
LinkTarget	Defines the way a link acts when a link is clicked.	String	<p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>
MaxResults	Maximum number of summaries displayed on a Web page.	Integer	<p>any integer</p> <p>0 = unlimited</p>
OrderKey	Sort the list by one of the values.	must be one of the values	<p>Title</p> <p>DateModified</p> <p>DateCreated</p> <p>LastEditorFname</p> <p>LastEditorLname</p> <p>See Also: "OrderKey Property" on page 140</p>
Random	Set to True if you want to randomly display one content block in the specified folder. The content changes each time a user views the page.	Boolean	<p>True</p> <p>False</p>
Recursive	Determines if the display includes content blocks in child folders of the selected folder.	Boolean	<p>True</p> <p>False</p>

Property	Description	Data Type	Values
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean	True False
WrapTag	Allows a developer to specify a server control's tag. The default is Span .	String	Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.

OrderKey Property

The OrderKey property determines what information appears to the right of the hyperlinked title. The values are described below.

Value	Description	Web Page View
DateModified	Displays the date the content was last modified.	<p>, RC Sportster. The RC Sportster is design mid level RC racer currently on the market</p> <p>ference (8/27/2002 2:44:44 PM) growing company in the RC host the 3rd annual RC conference in Atl</p>
DateCreated	Displays the date when the content was first created.	<p>id level RC racer currently on the mark</p> <p>arence (4/25/2002 4:42:55 PM) owing company in the RC st the 3rd annual RC conference in A</p>

Value	Description	Web Page View
LastEditorFname	Displays the first name of the user who last edited the content block.	mid level RC racer currently ference (John) growing company in the R host the 3rd annual RC cc
LastEditorLname	Displays the last name of the user who last edited the content block.	ct, KC Sportster. The KC Sp d mid level RC racer currentl Conference (Doe) st growing company in the R to host the 3rd annual RC cc

List Summary Example

Below is an example of the ListSummary server control properties.

Ektron	
Authenticated	False
ContentParameter	
ContentType	Content
Direction	Ascending
DisplayXslt	ecmNavigation
FolderID	14 
GetHtml	False
Hide	False
Language	0
MaxResults	0
OrderKey	Title
Random	False
Recursive	False
WrapTag	Span

The example has the following properties:

- Content Type - The results are a list of content blocks.
- Direction - The results are displayed in ascending order by title
- DisplayXslt - will use the navigation display format

- FolderID - Displays summaries from the folder with the ID 14
- GetHtml - Does not display the HTML from the content blocks in the list summary
- MaxResults - Unlimited results appear
- OrderKey - Summaries sorted by content block title
- Random - Displays summary for all content blocks in folder, not just one random one
- Recursive - Includes only selected folder, not its child folders

Retrieving the XML Structure of a List Summary

Retrieving the XML structure of XML content allows for greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a ListSummary server control onto it.
3. Set the FolderID property.
4. Drag and drop a Textbox on the Web form.
5. Set the TextMode property to MultiLine.



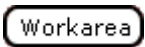



NOTE It is also recommended that you set the width of the text box to at least 400px.

6. On the code behind page, add the following line.
`Textbox1.Text = ListSummary1.XmlDoc.InnerXml`
7. Build the project.
8. View the Web form in a browser.
9. The XML structure of the list summary appears in the textbox.

Login Server Control

The Login server control paints a login button on the template when displayed in a browser.

When the Login server control is inserted and the project is built, the control displays the following buttons on a Web page.

Button	Description
 Login	When user is not logged in, this button appears. Clicking the button opens the login window, where a user can enter a username and password. Upon authentication, the user is logged in to the Ektron CMS400 Web site.
 Logout	After a user logs in, this button replaces the login button to let the user log out.
 Workarea	When logged in, this button appears under the logout button, allowing the user to access the Workarea.
 Preview	Lets the user preview the entire Web site as if all checked-in content were published. For more information, see the Ektron CMS400.NET User Manual section "Logging In and Out" > "Site Preview."
 XPreview	Turns off site preview mode.
 Help	Launches online, context sensitive help for Ektron CMS400.NET .

This subsection contains the following topics:

- ["Placing a Login Button" on page 143](#)
- ["Login Server Control Properties" on page 144](#)

Placing a Login Button

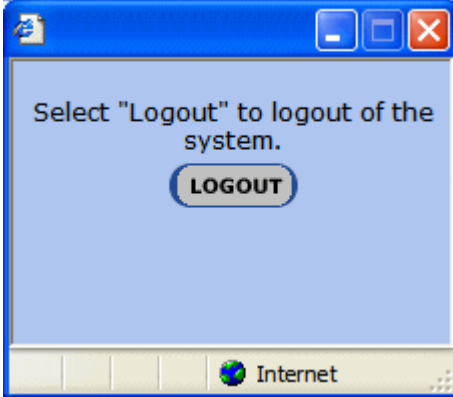


You can add any number of login buttons to a template. You can insert a login button on each template, or set up a special Web page, called login.aspx, from which users can log into the CMS400 site without the public being able to access the page.

Login Server Control Properties

The following table explains the properties of the Login server control.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
AutoLogin	<p>When set to true, users are automatically logged in using Active Directory authentication. They do not need to enter a username or password. The default is False.</p> <p>True = Use Active Directory authentication when logging in.</p> <p>False = Do not use Active Directory authentication when logging in.</p> <hr/> <p>Important!: For this property to function properly, you must be using Active Directory authentication with your CMS400.NET Site.</p> <hr/>	Boolean
Hide	<p>Used to hide login button in design time and run time.</p> <p>True = Hide login button</p> <p>False = Show login button</p>	Boolean
Language	Set a language for the Login server control. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
OnlyAllowMemberLogin	<p>This changes the Login button so only membership users can log in. This prevents someone from trying to login as an Ektron CMS400.NET user and gain access to the Ektron CMS400.NET Workarea. The default is False.</p> <p>True = Allow only membership users to login.</p> <p>False = Allow Ektron CMS400.NET users and membership users to login.</p>	Boolean

Property	Value	Data Type
PromptLogout	<p>When set to False, the logout process will not include the Logout Prompt window (shown below).</p>  <p>True = Users must click the Logout button in the Logout prompt window before they are logged out. False = Users are logged out without having to use the Logout prompt window.</p>	Boolean
SuppressHelpButton	<p>Hides the Help button that appears below the Login button when set to true. The default is False.</p> <p>True = Do not display the Help button.</p>  <p>False = Display Help button.</p> 	Boolean
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.</p>	String

Membership Server Control

The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist.

Below is an example of the Membership server control when viewed on a Web page.

The screenshot displays a web form for user registration or profile updates. It is divided into two main sections: 'Standard User Properties' and 'Custom User Properties'.

Standard User Properties (Red box):

- Username:
- First Name:
- Last Name:
- Password:
- Confirm Pwd:
- E-Mail Address:

Custom User Properties (Blue box):

- Street Address
- Zip Code
- Phone
- Birthday
- Music News
 - ☐ Country & Western
 - ☐ Rap
 - ☐ Rock & Roll
 - ☐ Rythem & Blues
 - ☐ Soul

Notification will send in language: English (U.S.)

Register Reset

When you create custom user properties, they appear for both membership users and **Ektron CMS400.NET** users. If you create a property such as Employee ID number, the property appears when a membership user views this form on a web page. The Membership server control allows you to decide if want the custom user properties to appear when the form is viewed.

It is recommended that you review the following documentation when using the Membership server control.

- The Administration Manual section “Managing Users & User Groups”
- The Administration Manual section “Custom User Properties”
- The Administration Manual section “Web Alert Feature”

Understanding these topics along with using this server control extends the most functionality to your membership users.

Membership Server Control Properties

The following table explains the properties of the Membership server control.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
DisplayMode	<p>The type of membership form displayed. Choose from the following.</p> <ul style="list-style-type: none"> • UserRegistration - Lets a site visitor register as a membership user. • ResetPassword - Lets a membership user reset his password. • UnsubscribeSecured - Lets a membership user unsubscribe. He needs to enter his user-name and password. • UnsubscribeUnsecured - Lets a membership user unsubscribe by entering his membership username only. • AccountActivate - Let a site visitor activate his membership account by entering the ID number in the account verification email. <p>For more information, see "DisplayMode" on page 150.</p>	String

Property	Value	Data Type
Hide	Used to hide login button in design time and run time. True = Hide login button False = Show login button	Boolean
Language	Set a language for the server control. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
RedirectFailedURL	The URL where a membership user is sent if the registration fails. <ul style="list-style-type: none"> • If the page resides in the same folder as the registration page, enter the name of the page. For example, RegFailed.aspx. • If the redirect page is located in a subfolder, add the folder path. For example, members\RegFailed.aspx. 	String
RedirectSucessURL	The URL where a membership user is sent when the registration succeeds. <ul style="list-style-type: none"> • If the page resides in the same folder as the registration page, enter the name of the page. For example, RegSucceed.aspx. • If the redirect page is located in a subfolder, add the folder path. For example, members\RegSucceed.aspx. 	String
RegisterButtonImg	Lets you add an image in place of the text on the register button. Enter a path to the image. For example: http://www.ektron.com/buttons/registerbutton.gif If the image is located in the site root, you do not need to enter the full path. Only enter the subfolder path and image name. For example: /buttons/registerbutton.gif	String
RegisterButtonText	Text that appears on the Register button. The default is Register . If you use a register button image, you do not see this text.	String

Property	Value	Data Type
ResetButtonImg	<p>Lets you add an image in place of the text on the reset button. Enter a path to the image. For example:</p> <p><code>http://www.ektron.com/buttons/resetbutton.gif</code></p> <p>If the image is located in the site root, you do not need to enter the full path. Only enter the subfolder path and image name. For example:</p> <p><code>/buttons/resetbutton.gif</code></p>	String
ResetButtonText	Text that appears on the reset button. The default is Reset . If you use a reset button image, you do not see this text.	String
ShowExtended	<p>Decide if the Custom User Properties are available when using this control. The default setting is True.</p> <p>True = Show Custom User Properties</p> <p>False = Hide Custom User Properties</p> <hr/> <p>Note: For more information on Custom User Properties, see the Administration Manual section "Custom User Properties".</p> <hr/>	String
UserExistsMessage	The message that appears when a membership user already exists. The default message is: Username(email) already exists!	String
UserSuccessMessage	The message that appears when a membership user successfully registers. The default message is: You have registered successfully.	String
UserUpdateSucessMessage	The message that appears when a membership user successfully updates their information. The default message is: You have successfully updated your information.	String

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

DisplayMode

Lets a developer decide what type of membership form is added to the Web form.
The following table describes settings for the DisplayMode property.

Setting	Description	Example
UserRegistration	Lets a site visitor register as a membership user.	<p>Username: <input type="text"/></p> <p>First Name: <input type="text"/></p> <p>Last Name: <input type="text"/></p> <p>Password: <input type="password"/></p> <p>Confirm Pwd: <input type="password"/></p> <p>E-Mail Address: <input type="text"/></p> <p><input type="button" value="Register"/> <input type="button" value="Reset"/></p>
ResetPassword	Lets a membership user reset his password.	<p>Username: <input type="text"/></p> <p><input type="button" value="Register"/> <input type="button" value="Reset"/></p>

Setting	Description	Example
UnsubscribeSecured	Lets a membership user unsubscribe. The membership user needs to enter his username and password to unsubscribe.	<p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Register"/> <input type="button" value="Reset"/></p>
UnsubscribeUnsecured	Lets a membership user unsubscribe by entering his membership username only.	<p>Username: <input type="text"/></p> <p><input type="button" value="Register"/> <input type="button" value="Reset"/></p>
AccountActivate	Let a site visitor activate his membership account by entering the ID number sent him in the account verification email.	<p>Username: <input type="text"/></p> <p>Account ID: <input type="text"/></p> <p><input type="button" value="Register"/> <input type="button" value="Reset"/></p>

Menu Server Control

The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt property. Below is a menu display with the SampleMenu Xslt.

- Products
 - RC Cars
 - [RC Cheetah](#)
 - [RC Sportster](#)
 - RC Planes
 - [RC Lilly](#)
 - [RC Redstar](#)
 - [Products Page](#)
 - [Visit Ektron.com](#)

Before you can use the Menu server control, you must create a menu in the **Ektron CMS400.NET** work area. To learn about creating menus, see “Working with Menus” in the **Ektron CMS400.NET** User Manual.

This subsection contains the following topics:

- “Menu Server Control vs. DhtmlMenu Server Control” on page 152
- “Menu Server Control Properties” on page 153
- “Using DisplayXslt Samples” on page 154
- “Retrieving the XML Structure of a Menu” on page 158

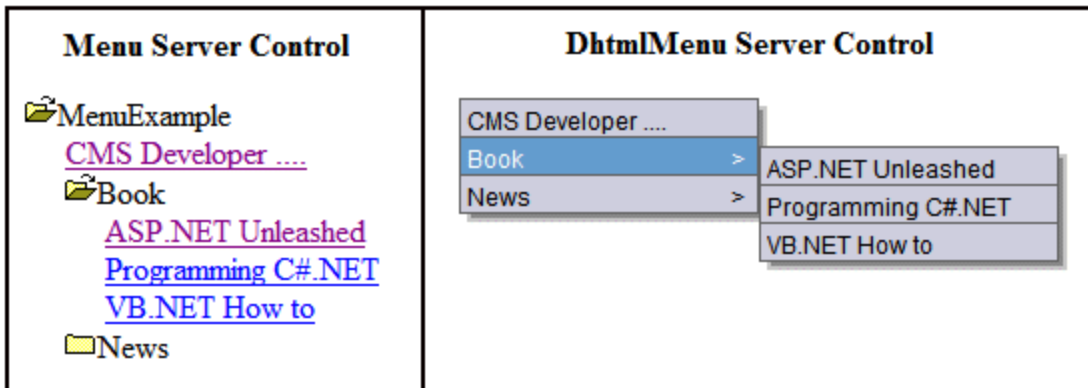
Menu Server Control vs. DhtmlMenu Server Control

The Menu server control calls a menu to be displayed on a page. You can manipulate the menu by using the DisplayXslt property.

With the DhtmlMenu server control, you display a dynamic menu with drop down boxes and submenus. You manipulate DhtmlMenus with a style sheet. DhtmlMenus save space on the Web page by having the drop down menus and submenus float over content.

WARNING!

You cannot create a DHTML menu using the code behind -- you must drag and drop it. Once the menu is on the template, you can manipulate its properties using the code behind.



Menu Server Control Properties

The Menu server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. <i>See Also:</i> "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). <i>See Also:</i> "Caching with Server Controls" on page 36.	Integer
DefaultMenuID	The ID of a menu that appears where you insert this server control if no other menu is identified or available. If you don't know the ID number of the menu, use the CMS Explorer to browse to it. <i>See Also:</i> "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	Integer

Property	Value	Data Type
DisplayXslt	<p>The Xslt to use to render the menu.</p> <p>None - databind only</p> <p>SampleMenu - A sample display, formatted as a bulleted menu list</p> <p>TreeMenu - A sample display, formatted as a folder tree. You can expand the tree by clicking on the folder icon.</p> <p>Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page</p> <p>For more information on using the SampleMenu and TreeMenu DisplayXslt, see "Using DisplayXslt Samples" on page 154.</p>	String
DynamicParameter	Select menu_id . When you do, this server control uses the menu passed as a URL parameter.	String
Hide	<p>Used to hide a menu in design time and run time.</p> <p>True = Hide menu</p> <p>False = Show menu</p>	Boolean
Language	Set a language for viewing menus. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Using DisplayXslt Samples

Two DisplayXslt samples are provided with the Menu server control, SampleMenu and TreeMenu. This section explains how to use them.

SampleMenu

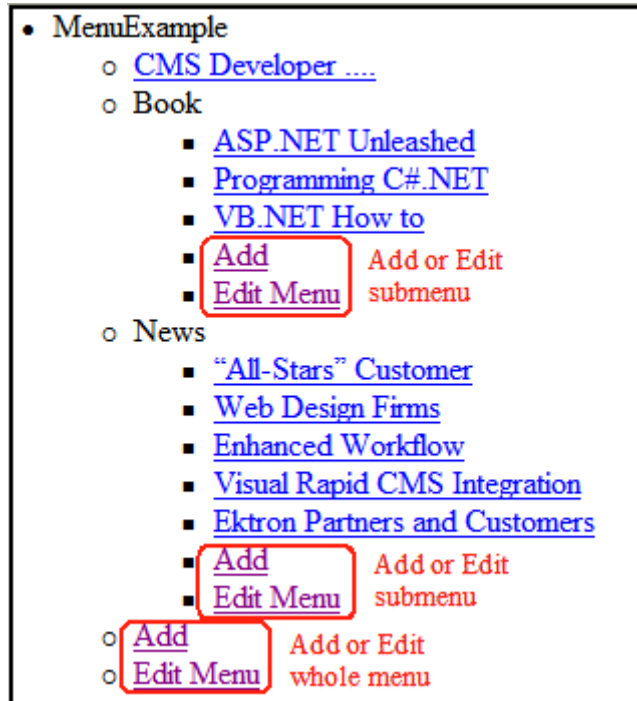
The SampleMenu DisplayXslt lets you display an **Ektron CMS400.NET** menu as a bulleted item list. See below for an example.

- MenuExample
 - [CMS Developer](#)
 - Book
 - [ASP.NET Unleashed](#)
 - [Programming C#.NET](#)
 - [VB.NET How to](#)
 - News
 - [“All-Stars” Customer](#)
 - [Web Design Firms](#)
 - [Enhanced Workflow](#)
 - [Visual Rapid CMS Integration](#)
 - [Ektron Partners and Customers](#)

When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each section of the menu: Add and Edit Menu. The user can use these to add a new menu item or edit an existing one. See example below.

For Information on a adding a menu Item, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Adding a New Content Block to a Menu via Navigation Link on a Web Page”.

For Information on a editing a menu, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Editing a Menu via Navigation Link on a Web Page”.



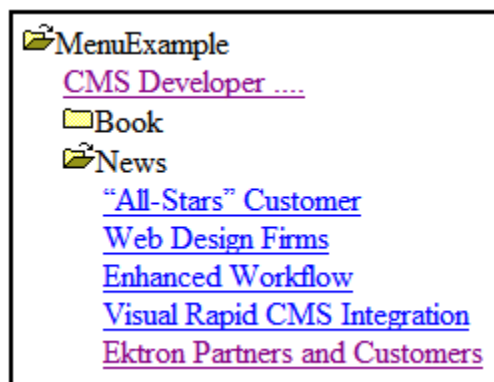
SampleMenu XSL code

For an example of the XSL code used in the SampleMenu, click the following link and choose the appropriate version.

<http://www.ektron.com/developers/cms400kb.cfm?id=2114>

TreeMenu

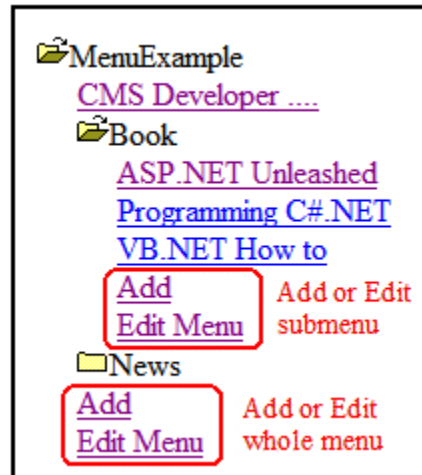
The TreeMenu DisplayXslt lets you display an **Ektron CMS400.NET** menu as a clickable folder list. You can expand and collapse the menu by clicking on the folder icons. See below for an example.



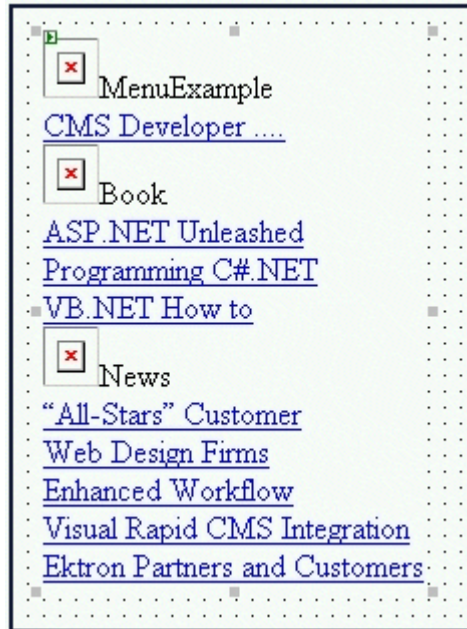
When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each section of the menu: Add and Edit Menu. The user can use these to add a new menu item, or edit an existing menu item. See example below.

For Information on a adding a menu Item, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Adding a New Content Block to a Menu via Navigation Link on a Web Page”.

For Information on a editing a menu, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Editing a Menu via Navigation Link on a Web Page”.



NOTE When using the TreeMenu server control with Visual Studio and Windows 2003, the folder images do not display during design-time. They do, however, display correctly during run-time.



TreeMenu XSL code

For an example of the XSL code used in the MenuTree, click the following link and choose the appropriate version.

<http://www.ektron.com/developers/cms400kb.cfm?id=2114>

Retrieving the XML Structure of a Menu

Retrieving the XML structure of XML content allows for greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a Menu server control onto it.
3. Set the DefaultMenuID properties.
4. Drag and drop a Textbox on the Web form.
5. Set the TextMode property to MultiLine.

NOTE It is also recommended that you set the width of the text box to at least 400px.

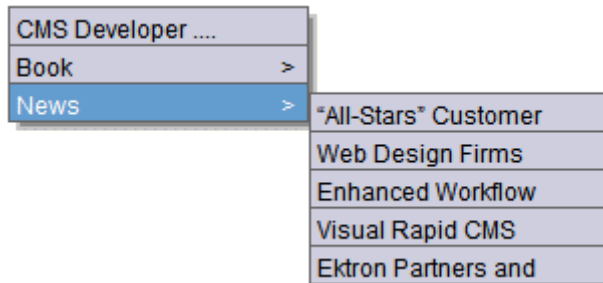
6. On the code behind page, add the following line.
`Textbox1.Text = Menu1.XmlDoc.InnerXml`
7. Build the project.

8. View the Web form in a browser.
9. The XML structure of the menu appears in the textbox.

DhtmlMenu Server Control

NOTE For optimal performance, Ektron recommends using at minimum Internet Explorer 5.5, Netscape 6.0, or Mozilla Firefox 1.0.

The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot of space. You can modify the menu's style via a style sheet.

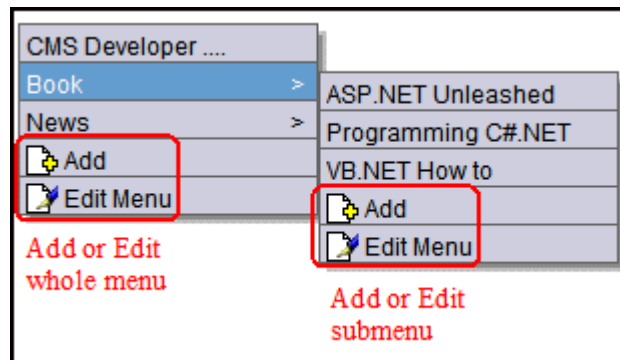


Before using the DhtmlMenu server control, you must add items to a menu in the Ektron CMS400.NET workarea. To learn about creating menus, see "Working with Menus" in the CMS400.NET User Manual.

The DhtmlMenu server control also lets developers add content to a menu. When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each menu: Add and Edit Menu. The user can use these to add a new menu item or edit an existing menu item. See example below.

For Information on adding a menu item, see **Ektron CMS400.NET** Users Manual > "Working with Menus" > "Managing Menus" > "Adding a New Content Block to a Menu via Navigation Link on a Web Page".

For Information on editing a menu, see **Ektron CMS400.NET** Users Manual > "Working with Menus" > "Managing Menus" > "Editing a Menu via Navigation Link on a Web Page".



This subsection contains the following topics:

- ["Menu Server Control vs. DhtmlMenu Server Control" on page 161](#)
- ["DhtmlMenu Server Control Properties" on page 161](#)
- ["Using a Style Sheet" on page 165](#)
- ["Formatting the DhtmlMenu Server Control" on page 166](#)

Menu Server Control vs. DhtmlMenu Server Control

See ["Menu Server Control vs. DhtmlMenu Server Control" on page 152.](#)


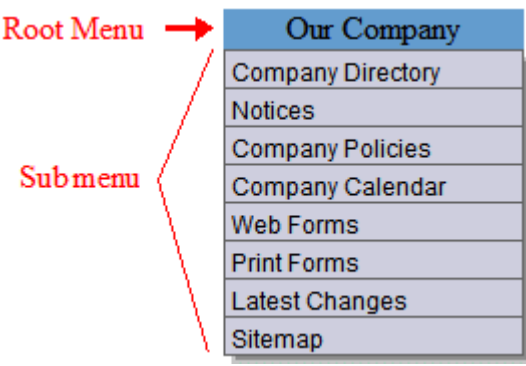
DhtmlMenu Server Control Properties


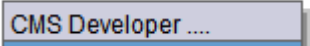
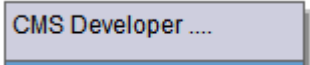
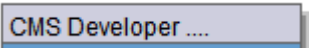
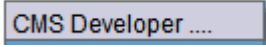
The DhtmlMenu server control properties are described in this table.

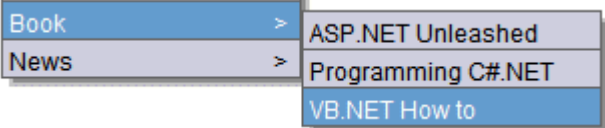
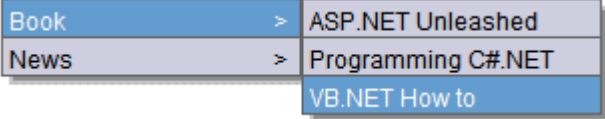
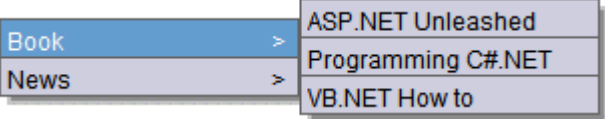
Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Align	Specifies the horizontal alignment of the root menu text. The default is Center . NotSet - Use the default setting. Left - Aligns text to the left. Center - Aligns text to the center. Right - Aligns text to the right. Justify - Aligns text justified to both margins.	String
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc.... See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
BackColor	Sets the background color for the root menu. This property appears in the Appearance category.	String

Property	Value	Data Type
CssClass	<p>Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet property. See Also: "Stylesheet" on page 164</p> <p>Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet property.</p> <p>For example, if you enter lowtext in the property, the root menu changes as shown below.</p> <div> <div>No CssClass applied</div> <div>lowText CssClass applied</div> </div> <div> <div>MenuExample</div> <div>MenuExample</div> </div>	String
DefaultMenuID	<p>The ID of a menu that appears where you inserted this server control if no other content block is identified, or is not available.</p> <p>If you don't know the ID number of the menu, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p>	Integer
DynamicParameter	Select menu_id . When you do, this server control uses the menu passed as a URL parameter.	String
ForeColor	Sets the color of the font for the root menu. This property appears in the Appearance category.	
Hide	<p>Used to hide a menu in design time and run time.</p> <p>True = Hide menu</p> <p>False = Show menu</p>	Boolean

Property	Value	Data Type
HorizontalDisplay	<p>Displays submenu to the right of the root menu. This only affects the first submenu of the root menu. All subsequent submenus appear to the right of each submenu.</p> <p>The default is False</p> <p>True = Submenu appears to the right of the root menu</p>  <p>False = Submenu appears below the root menu</p> 	Boolean
Language	Set a language for viewing Dhtml Menus. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
MenuHoverOverColor	Sets the background color for the root menu when hovering over it.	String

Property	Value	Data Type
MenuStyle	<p>A specified inline style that defines the root menu. For example, if you enter the following in the MenuStyle property box:</p> <pre>font: bold 18px Arial;color:royalblue; background-color:slateblue</pre> 	String
Stylesheet	<p>Specify the location and style sheet for the submenu. For example: \Workarea\csslib\mytest.css. Leave blank to use the default.</p> <p>pop_style.css is the default style sheet. It is located in \webroot\CMS400demo\Workarea\csslib. For more information on style sheets, see "Using a Style Sheet" on page 165.</p> <p>The Stylesheet property can be used in conjunction with the CssClass property to define the root menu. See Also: "CssClass" on page 162</p>	String
SubMenuBackColor	Sets the background color for the submenu.	String
SubMenuHoverOverColor	Sets the background color for the submenu when hovering over it.	String
SubMenuItemHeight	<p>Sets the height of each submenu item in pixels. The Default is 20</p> <p>Example set to 20: </p> <p>Example set to 30: </p>	Integer
SubMenuItemWidth	<p>Sets the width of each submenu item in pixels. The Default is 150</p> <p>Example set to 150: </p> <p>Example set to 130: </p>	Integer

Property	Value	Data Type
SubMenuTopOffset	<p>Sets the location of the top of the submenu. This number is in pixels. The default is 5</p> <p>Example set to 5:</p>  <p>Example set to 0 (zero):</p>  <p>Example set to -10:</p> 	Integer
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Using a Style Sheet

You can create your own style sheet, or modify the existing default one, `pop_style.css`. It is located in `\webroot\CMS400demo\Workarea\csslib`. For a description of the classes contained in `pop_style.css`, see "[pop_style.css](#)" on page 644.

Style Sheets let you choose different text colors, text alignment, text fonts, and text size to create a customized menu.

Best Practice

Ektron recommends copying and renaming the default style sheet. This insures you always have a clean file to start with. This also gives you something to reference if you are not getting the results you expect.

The style sheet must be located in the site root.

You can only define one menu style sheet for each Web page. If you have more than one menu on a single Web page, they share the style sheet.

For more information on style sheets, visit <http://www.w3.org/Style/CSS/>.

Formatting the DhtmlMenu Server Control

Additional information on formatting the DhtmlMenu server control can be found in the following Knowledge Base article.

http://dev.ektron.com/kb_article.aspx?id=2441

MetaData Server Control

The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.

This subsection contains the following topics:

- ["MetaData Server Control vs. MetaDataList Server Control" on page 167](#)
- ["MetaData Server Control Properties" on page 167](#)
- ["Using the Simple Dublin Core Metadata Standard" on page 168](#)
- ["Using the MetaData Server Control" on page 169](#)

MetaData Server Control vs. MetaDataList Server Control

With the MetaData server control, you add metadata from content blocks to your Web page. With the MetaDataList server control, you create a list of content blocks to display on your site, based on the Metadata in each content block. For Information on the MetaDataList server control, see ["MetaDataList Server Control" on page 171](#).

MetaData Server Control Properties

The Metadata server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String

Property	Value	Data Type
DefaultContentID	<p>The ID of the content block that the server control will get the metadata from.</p> <p>If you want to add metadata from several content blocks, set this property to zero (0) and use the DefaultItemList property to identify the content blocks.</p> <p>If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p>	Integer
DefaultItemList	<p>A comma delimited list of content block IDs to get metadata from. DefaultContentID must be set to zero (0) so you can add IDs to the list.</p> <p><code>DefaultItemList [12,4,7]</code></p>	Integer
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.	String
GenerateDublinCore	<p>When enabled, this property automatically creates seven of the Simple Dublin Core metadata fields from standard CMS400.NET system properties. The default is false.</p> <p>True = Generate Simple Dublin Core metadata fields</p> <p>False = Do not generate Simple Dublin Core metadata fields</p> <p>The seven fields and how they are associated with the CMS400.NET properties is explained in "Using the Simple Dublin Core Metadata Standard" on page 168</p>	Boolean
Hide	<p>Used to hide a Metadata server control in design time and run time.</p> <p>True = Hide Metadata server control</p> <p>False = Show Metadata server control</p>	boolean
Language	Set a language for viewing form content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer

Using the Simple Dublin Core Metadata Standard

Simple Dublin Core is a set of fifteen standard names for metadata fields designed to cover the most useful items of information on a document. From the Dublin Core FAQ: Dublin Core metadata provides card catalog-like definitions for defining the properties of objects for Web-based resource discovery systems.

For more information on the Dublin Core standard refer to the Usage Guide: <http://www.dublincore.org/documents/usageguide/>

To generate Dublin Core metadata, set the GenerateDublinCore property to True. This creates seven of the fifteen Dublin Core metadata fields. These fields are automatically filled with the information from the equivalent Ektron CMS400.NET property. Below is a list of the seven fields that are created and their Ektron CMS400.NET equivalent property.

IMPORTANT!

These seven Dublin Core fields are automatically populated with information from their equivalent Ektron CMS400.NET property.

Dublin Core Field Name	Ektron CMS400.NET Property
DC.title	Content block title
DC.description	Plain text version of a content block teaser (summary)
DC.contributor	Content block last editor name
DC.date	Content block last edit date
DC.format	"text/html"
DC.identifier	URL of current page (from ASP.NET Server.Request object)
DC.language	CMS language cookie / current site language, expressed as a .NET System.Globalization Culture Name

To fully comply with the Simple Dublin Core metadata element set, the administrator must create the remaining eight Dublin Core fields as standard CMS400.NET Metadata fields and apply them to all CMS400.NET folders. Next, the CMS users must fill in the appropriate values for each content block.

For information on the remaining eight Simple Dublin Core fields, see the Administrator Manual section "Managing Content Folders" > "Metadata" > "Simple Dublin Core Metadata".

Using the MetaData Server Control

These steps show how to use the MetaData server control.

1. Drag a MetaData server control into a template.
2. Set the properties of the Metadata server control. This will create the following HTML in the HTML body.

```
<cms:MetaData id="MetaData1" runat="server" DefaultContentID="12"></cms:MetaData>
```

Or if you are using multiple content block IDs In the DefaultItemList, the following HTML will be created.

```
<cms:metadata id="MetaData1" runat="server" DefaultItemList="[12,7,4]"></cms:metadata>
```

3. Click on the HTML tab and copy that line from the <body> tag into the <head> tag.

```
<HEAD>
<title>WebForm5</title>
<cms:MetaData id="Metadata2" runat="server" DefaultContentID="12"></cms:MetaData>
<meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
<meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/i
</HEAD>
<body>
  <form id="Form1" method="post" runat="server">|
  </form>
</body>
```

4. Save the Web form and rebuild the solution.
5. Open the Web page in the browser.
6. Right click on the Web page and click **View Source**. Look in the head tag. The meta tags from the content block are added to the page.

The metadata information added is shown below.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>WebForm5</title>
    <title>CMS Developer</title>
    <meta name="keywords" content="CMS400; .NET; CMS">
    <meta name="description" content="CMS400 developer section">
    <meta content="Microsoft visual studio .NET 7.1" name="GENERATOR">
    <meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
```

MetaDataList Server Control

Use the MetaDataList server control to create lists based on Keyword Names and Keyword Values contained within the metadata of a content block. In the example below the KeyWordNames are circled in red and the KeyWordValues are circled in green.

Title: [English (U.S.)]

Content **Summary** **Metadata** **Schedule** **Comment**

* = Required fields

Title: (500 max.) Separator Character ";"

486 characters left

Keywords: (500 max.) Separator Character ";"

The list can display the information as a list of hyperlinks. You can choose, based on properties you set, if you would like to have a summary displayed. You can also choose how to order the display. For information on adding Metadata to a content block, see the Users Manual > "Adding or Editing Metadata".

This subsection contains the following topics:

- ["MetaData Server Control vs. MetaDataList Server Control" on page 171](#)
- ["MetaDataList Server Control Properties" on page 171](#)
- ["Retrieving the XML Structure of a MetaDataList" on page 175](#)

MetaData Server Control vs. MetaDataList Server Control

With the MetaData server control, you add metadata from content blocks to your Web page. With the MetaDataList server control, you create a list of content blocks to display on your site based on the Metadata in each content block. For Information on the MetaData server control, see ["MetaData Server Control" on page 167](#)

MetaDataList Server Control Properties

The MetaDataList server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36	Integer
ContentType	One of the following <ul style="list-style-type: none"> all types archived content archived forms content forms To learn about archived content, see the Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting Archive Options."	String
DisplayXslt	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every content block in the folder See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists the title of every content block in the folder plus the content summary. See Also: "Example of ecmTeaser Display" on page 85 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page	String

Property	Value	Data Type
ExactPhrase	<p>Determines whether the KeyWordValue needs to match the phrase exactly.</p> <p>For example, if "site" is the KeyWordValue and the title of a content block is "Welcome to the site" and ExactPhrase is set to true, you would not see the content block in the metadata list. This is because "site" does not equal "Welcome to the site".</p> <p>True = Match the exact phrase False = Doesn't need to match exact phrase</p>	Boolean
FolderID	The root folder content will be pulled from.	Integer
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the collection. For example, display content inside a web server control such as a GridView.	Boolean
Hide	<p>Used to hide a metadata list in design time and run time.</p> <p>True = Hide metadata list False = Show metadata list</p>	Boolean
KeywordName	<p>The KeywordName is the container where the KeywordValues are located.</p> <p>Some examples of a KeywordName might be "Keywords or Title"</p> <p>For information on creating KeywordNames for Metadata, see the Ektron CMS400.NET Administrators Manual > "Managing Content" > "Metadata" > "Managing Metadata Definitions" > "Adding a Metadata Definitions"</p>	String
KeywordValue	<p>The value that is associated with the KeywordName</p> <p>Some examples of a KeywordValue might be "Home; page; or company"</p>	String
Language	Set a language for viewing the MetaDataList. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer

Property	Value	Data Type
LinkTarget	<p>Defines the way a link acts when a link is clicked. Choices are:</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>	String
MaxNumber	<p>The maximum number of results returned for display.</p> <p>0 = unlimited</p>	Integer
OrderBy	<p>The order of the list to be returned.</p> <ul style="list-style-type: none"> • Title - The title of the content block • ID - The content block ID number • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • LastEditorLname - The last editor’s last name • LastEditorFname - The last editor’s first name 	String
Recursive	<p>Whether to search sub-folders of the identified root folder. The root folder is identified in the FolderID property.</p>	Boolean
SortOrder	<p>Ascending or Descending</p>	String

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Retrieving the XML Structure of a MetadataList

Retrieving the XML structure of XML content allows for greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a MetadataList server control onto it.
3. Set the KeywordName and KeywordValue properties.
4. Drag and drop a Textbox on the Web form.
5. Set the TextMode property to MultiLine.

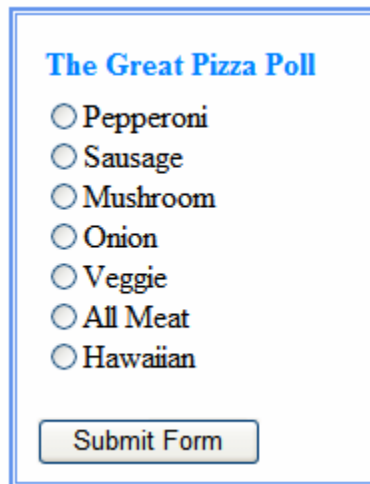
NOTE It is also recommended that you set the width of the text box to at least 400px.

6. On the code behind page, add the following line.

```
Textbox1.Text = Metadata1.XmlDoc.InnerXml
```
7. Build the project.
8. View the Web form in a browser.
9. The XML structure of the MetadataList appears in the textbox.

Poll Server Control

The Poll server control displays a poll or survey created from an Ektron CMS400.NET form on a Web page. When added to a template and visited, the poll might look like this. You can change a poll's appearance by modifying its properties.

A screenshot of a web form titled "The Great Pizza Poll" in blue text. Below the title is a list of seven radio button options: Pepperoni, Sausage, Mushroom, Onion, Veggie, All Meat, and Hawaiian. At the bottom of the form is a "Submit Form" button.

While you can use a FormBlock server control to add a poll or survey, Ektron recommends using a poll server control, because it provides greater flexibility on the poll's appearance.

Typically, developers or site administrators want a poll or survey to be a small part of a Web page not the main content. By using the EnableAjax property you can display the results in the same area as the poll or survey without disturbing the surrounding content.

For a detailed description of these properties, click on them or see ["Poll Server Control Properties" on page 176](#).

This subsection contains the following topics:

- ["Poll Server Control Properties" on page 176](#)
- ["Automatic versus Manual Generation of Form Tags" on page 178](#)

Poll Server Control Properties

The following table explains the properties of the poll server control.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
(Poll ID)	The ID of the poll that appears where you inserted this server control. If you don't know the ID number of the form block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.	Integer
AddValidation	Set to True to add Validation to your main runat=server form.	Boolean
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15.	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.	Integer
DynamicParameter	To make this form block dynamic, select id . When you do, this server control uses the form block passed as a URL parameter.	String
EnableAjax	Displays the poll or results, using an <code>iFrame</code> , in the area of the page that contains the poll without disturbing its surrounding contents. The default is true. True = Polls and results are shown in an <code>iFrame</code> without any modification or refresh of the surrounding contents. False = The poll and the results replace the Web page's content.	
Hide	Used to hide a form block in design time and run time. True = Hide form block False = Show form block	Boolean
Include Tags	Determines if tags are generated automatically of manually. See Also: "Automatic versus Manual Generation of Form Tags" on page 178	Boolean

Property	Value	Data Type
Language	Set a language for viewing form content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Automatic versus Manual Generation of Form Tags

When using the poll server control's Include tags property, you have two options.

Option	Recommended if you are developing	How to Enable
Let Ektron CMS400.NET generate form tags automatically	A pure script .aspx page	Set Include Tags property to True .
Modify HTML form tags in the .aspx file	An .aspx page and associated code-behind web form	<p>Set Include Tags property to False.</p> <p>Here is the default .NET generated form tag:</p> <pre><form id="Form1" method="post" runat="server"></pre> <p>Modify the form tag as indicated in red:</p> <pre><form id="Form1" method="post" runat="server" OnSubmit="EkFmValidate(this);"></pre>

RssAggregator Server Control

An RSS Aggregator or reader takes an RSS feed and displays the information in a readable format. The RssAggregator server control lets you do the same by processing and displaying an RSS feed from any Web site. This allows you to create an information Web page for news, stories, images, lists of music, or for whatever purpose someone creates an RSS feed.

You can create a multi-level information Web page by using multiple RssAggregator server controls on the same page. For example, you could create a world news Web page by adding RSS feeds from the NY Times, The BBC, and AFP (Agence France-Presse).

The process of adding an RSS Aggregator to your Web site is simple.

1. Choose a Web form where the RSS Aggregator will reside
2. Drag and drop the RssAggregator server control on that Web form
3. Set the URL property to point at the RSS feed. For example, `http://msdn.microsoft.com/rss.xml`
4. Save the Web form

When a visitor to your site views the Web form in a browser, he sees the RSS feed displayed properly. When he refreshes the Web page, any information that the RSS feed provider has updated is now reflected. See Also: ["Using the RssAggregator Server Control" on page 181](#)

By creating a custom XSLT, you can change the way an RSS feed is viewed by your site visitors.

This subsection contains the following topics:

- ["RSSAggregator Server Control Properties" on page 179](#)
- ["Using the RssAggregator Server Control" on page 181](#)

RSSAggregator Server Control Properties

The RSSAggregator server control properties are described in this table.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
DisplayXslt	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every RSS feed item See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists a title and a description of every RSS feed item See Also: "Example of ecmTeaser Display" on page 85 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page	String
Hide	Used to hide an RSS Aggregator server control in design time and run time. True = Hide Metadata server control False = Show Metadata server control	boolean
Language	Set a language for viewing the RssAggregator. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
MaxResults	The Maximum number of items from an RSS feed that are returned (0=unlimited).	Integer
URL	The RSS feed path for the server control. For example: <code>http://msdn.microsoft.com/rss.xml</code>	String
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Using the RssAggregator Server Control

These steps show how to use the RssAggregator server control.

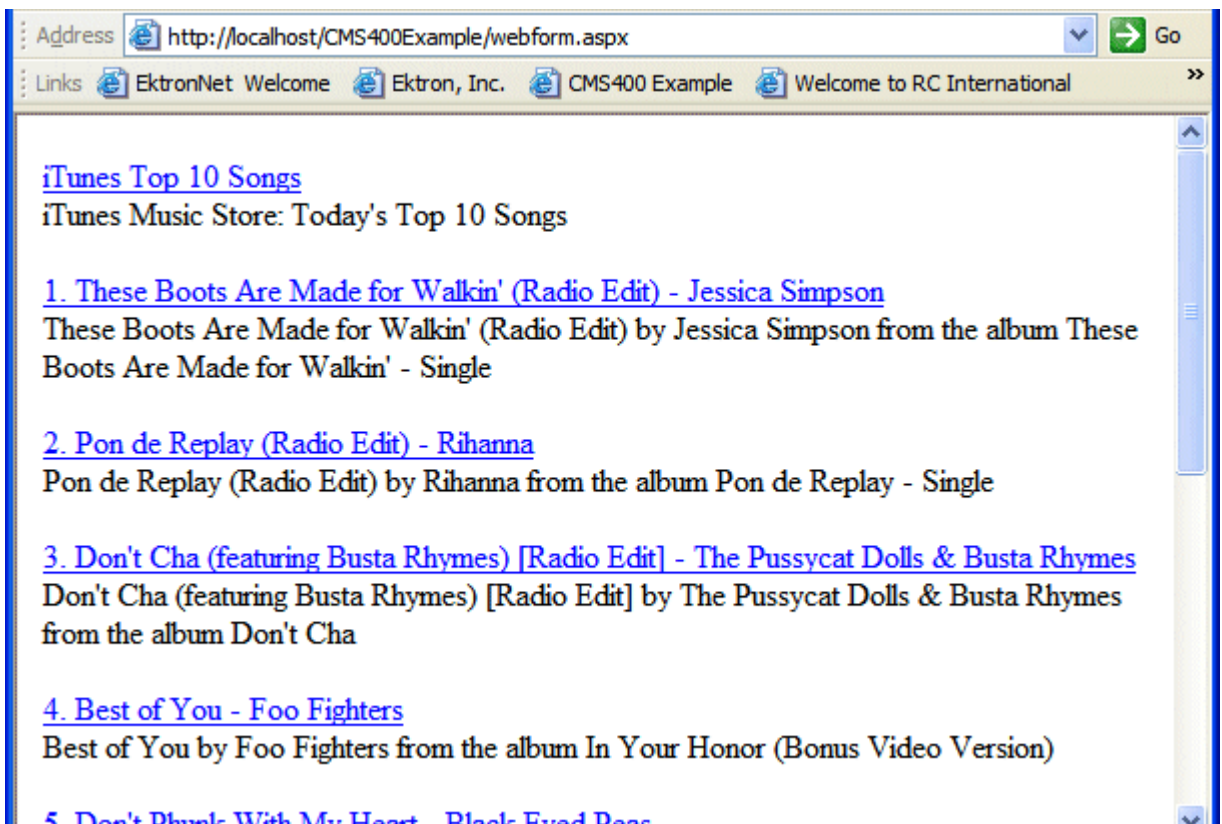
1. Drag an RssAggregator server control into a template.
2. Set the URL property with the path of the RSS feed you want to display. This example use the following RSS feed:

```
http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStore.woa/wpa/MRSS/topsongs/limit=10/rss.xml
```

This feed displays the top 10 songs according to the iTunes Music Store.

NOTE In addition to the URL property, other Ektron properties are available to customize this server control. For more information on these properties, see ["RSSAggregator Server Control Properties" on page 179](#)

3. Build the Web form.
4. Browse to the Web page in your browser.


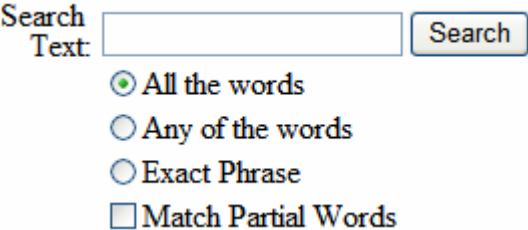
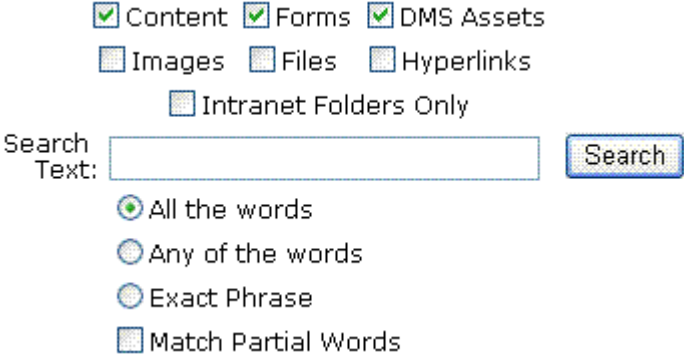


5. The page appears with an RSS list of information.

Search Server Control

Introduction

The Search server control displays a text box with a search button on the template. The search is highly customizable and allows you to provide a basic search, a standard search, or advanced search. The different types of searches are explained in ["Using the Advanced Search Properties" on page 206](#). Below is a visual example of the different searches.

Description	Web Page View
A Basic Search	
A Standard Search	
An Advanced or Mixed Search	

When users perform a search using the Search server control, they can be looking through Ektron CMS400.NET content, DMS assets, forms and library items. To learn more about configuring the Search server control to search

different content types, see ["Using the Advanced Search Properties" on page 206](#).

Searches can also be performed on content that utilizes the Searchable Property metadata type. You can make this search parameter available to users performing a search or hide it. See [Also: "ShowExtendedSearch" on page 202](#)

To search XML content (that is, content created with Smart Forms), use the IndexSearch server control. See [Also: "Search Server Control vs. IndexSearch Server Control" on page 183](#).

This subsection contains the following topics.

- ["Search Server Control vs. IndexSearch Server Control" on page 183](#)
- ["Search Server Control Properties" on page 183](#)
- ["Property Usage Table" on page 184](#)
- ["Property Descriptions" on page 185](#)
- ["Text Box Examples" on page 199](#)
- ["Button Image Source Examples" on page 199](#)
- ["Search Button Text Examples" on page 200](#)
- ["Display Examples" on page 201](#)
- ["ShowDate Examples" on page 202](#)
- ["ShowExtendedSearch" on page 202](#)
- ["Using the Advanced Search Properties" on page 206](#)
- ["Fieldset Properties" on page 212](#)
- ["Retrieving the XML Structure of a Search" on page 213](#)
- ["Displaying Search Results on a Separate Page" on page 214](#)
- ["Using a Style Sheet" on page 216](#)
- ["Formatting the Search Server Control" on page 217](#)

Search Server Control vs. IndexSearch Server Control

The Search server control lets users search your Web site for HTML Content (Both active and archived), DMS assets, forms and library items. The IndexSearch server control lets users search XML content on your Web site. For more information on the IndexSearch server control, see ["IndexSearch Server Control" on page 218](#).

Search Server Control Properties

The following tables explain the properties of the Search server control. The first table shows whether the properties affect the search display, the search criteria or the search results display. The second table contains a list of all the properties,

their descriptions, data type and links to more information. The rest of the tables provide further information on the various Search server control properties.

Property Usage Table

Properties in the Search server control generally affect the control in one of three ways:

- **Site Search Display** - the Search server control's appearance on your site
- **Site Search Criteria** - the criteria the control uses to search the site
- **Search Results Display** - how the search results are displayed on your site

The following table shows which properties are associated with each category.
See Also: ["Property Descriptions"](#) on page 185

Site Search Display	Site Search Criteria	Search Results Display
<ul style="list-style-type: none"> • ButtonImgSrc • ButtonText • ContentFieldsetLegend • Display • EnableAdvancedLink • EnableArchivedBtn • EnableContentBtn • EnableContentFieldset • EnableDMSAssetsBtn • EnableFolderFieldset • EnableFormsBtn • EnableLibFilesBtn • EnableLibHyperlinksBtn • EnableLibImageBtn • EnableLibraryFieldset • FolderFieldsetLegend • Hide • Language • LibraryFieldsetLegend • MaxCharacters • ShowExtendedSearch • ShowSearchOptions • Text Box Size • WrapTag 	<ul style="list-style-type: none"> • EnableArchived • EnableContent • EnableDMSAssets • EnableForms • EnableLibFiles • EnableLibHyperlinks • EnableLibImages • FolderID • Language • Recursive • ShowSearchOptions • TargetFolderDescription-List 	<ul style="list-style-type: none"> • DisplayXslt • EmptyResultMsg • Hide • Language • LinkTarget • MaxResults • MaxTeaserLength • OrderBy • OrderDirection • PostResultsAt • RemoveTeaserHtml • ResultTagId • ShowDate • ShowSearchBoxAlways

Property Descriptions

The following table describes the Search server control Properties. See Also: ["Property Usage Table"](#) on page 184

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Description	Data Type	For more information, see
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderID property.	String	"Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15
ButtonImgSrc	If you want to display an image on the submit button, enter the server path to it.	String	"Button Image Source Examples" on page 199
ButtonText	The text used for the button if no image source is identified. If an image source is identified, this is alternative text for the button.	String	"Search Button Text Examples" on page 200
ContentFieldsetLegend	Text that appears in the fieldset border around the content check boxes. This property is used only when the EnableContentFieldset property is set to True. The default is: Content Items:	String	"Fieldset Properties" on page 212 See Also: "EnableContentFieldset" on page 188
Display	Determines if the search options are in a row (Horizontal) or a column (Vertical).	String	"Display Examples" on page 201
DisplayXslt	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every content block in the folder ecmTeaser - lists the title of every content block in the folder plus the content summary Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page	String	ecmNavigation - "Example of ecmNavigation Display" on page 85 ecmTeaser - "Example of ecmTeaser Display" on page 85

Property	Description	Data Type	For more information, see
EmptyResultMsg	The return message displayed if the search returns no hits. The default is: Search returned zero results	String	
EnableAdvancedLink	Set to true to display the link that makes the advanced properties available. True = Display Advanced link False = Hide Advanced link	Boolean	"Using the Advanced Search Properties" on page 206
EnableAjax	Set to true to enable Ajax searches. When enabled, the <code>MaxResults</code> property determines the maximum number of results per page, and the <code>ResultTagId</code> property becomes active. True = Enable Ajax Search False = Original HTML Search	Boolean	"Ajax Search" on page 197 See Also: "MaxResults" on page 193 and "ResultTagId" on page 195.
EnableArchived	Enables users to search archived content. The archived content must be archived as Archived Remain On Site. The default is False. True = Searching archived content is enabled. False = Searching archived content is disabled, unless Include Archived is checked in the Advanced search. For more information on archiving content, see the User Manual section "Scheduling Content to Begin and End" > "Setting Archive Options".	Boolean	"Using the Advanced Search Properties" on page 206 See Also: "EnableArchivedBtn" on page 188

Property	Description	Data Type	For more information, see
EnableArchivedBtn	<p>Provides a check box so users can decide if they want to perform an archive search. The default is False.</p> <p>True = The Include Archive check box is enabled in the advanced search properties. If used with EnableArchived, the check box is checked by default.</p> <p>False = The Include Archive check box is disabled in the advanced search properties.</p>	Boolean	"Using the Advanced Search Properties" on page 206
EnableContent	<p>Enables users to search content. The default is True.</p> <p>True = Searching content is enabled.</p> <p>False = Searching content is disabled, unless Content is checked in the Advanced search.</p>	Boolean	"Using the Advanced Search Properties" on page 206 See Also: "EnableContentBtn" on page 188
EnableContentBtn	<p>Provides a check box so users can decide if they want to perform a content search. The default is True.</p> <p>True = The Content check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p>False = The Content check box is disabled in the advanced search properties.</p>	Boolean	"Using the Advanced Search Properties" on page 206
EnableContentFieldset	<p>Creates a border around the group of content items in an advanced search. These content items include Content, Forms and DMS Assets.</p> <p>The default is False.</p> <p>True = Show border around content items.</p> <p>False = No border around content items.</p>	Boolean	"Fieldset Properties" on page 212 See Also: "ContentFieldsetLegend" on page 186

Property	Description	Data Type	For more information, see
EnableDMSAssets	<p>Enables users to search DMS Assets. The default is True.</p> <p>True = Searching DMS Assets is enabled.</p> <p>False = Searching DMS Assets is disabled, unless DMS Assets is checked in the Advanced search.</p>	Boolean	<p>"Using the Advanced Search Properties" on page 206</p> <p>See Also: "EnableDMSAssetsBtn" on page 189</p>
EnableDMSAssetsBtn	<p>Provides a check box so users can decide if they want to perform a DMS Asset search. The default is True.</p> <p>True = The DMS Asset check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p>False = The DMS Asset check box is disabled in the advanced search properties.</p>	Boolean	<p>"Using the Advanced Search Properties" on page 206</p>
EnableFolderFieldset	<p>Creates a border around the group of folders in an advanced search. These folders include any folders listed in the TargetFolderDescriptionList property. The default is False.</p> <p>True = Show border around folders.</p> <p>False = No border around folders.</p>	Boolean	<p>"Fieldset Properties" on page 212</p> <p>See Also: "FolderFieldsetLegend" on page 192 and "TargetFolderDescriptionList" on page 197</p>
EnableForms	<p>Enables users to search Forms. The default is True.</p> <p>True = Searching Forms is enabled.</p> <p>False = Searching Forms is disabled, unless Forms is checked in the Advanced search.</p>	Boolean	<p>"Using the Advanced Search Properties" on page 206</p> <p>See Also: "EnableFormsBtn" on page 190</p>

Property	Description	Data Type	For more information, see
EnableFormsBtn	<p>Provides a check box so users can decide if they want to perform a Forms search. The default is True.</p> <p>True = The Forms check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p>False = The Forms check box is disabled in the advanced search properties.</p>	Boolean	"Using the Advanced Search Properties" on page 206
EnableLibFiles	<p>Enables users to search Library Files. The default is True.</p> <p>True = Searching Library Files is enabled.</p> <p>False = Searching Library Files is disabled, unless Files is checked in the Advanced search.</p>	Boolean	"Using the Advanced Search Properties" on page 206 See Also: "EnableLibFilesBtn" on page 190
EnableLibFilesBtn	<p>Provides a check box so users can decide if they want to perform a Library Files search. The default is True.</p> <p>True = The Files check box is enabled in the advanced search properties. If used with EnableLibFiles, the check box is checked by default.</p> <p>False = The Files check box is disabled in the advanced search properties.</p>	Boolean	"Using the Advanced Search Properties" on page 206
EnableLibHyperlinks	<p>Enables users to search Library Hyperlinks. The default is True.</p> <p>True = Searching Library Hyperlinks is enabled.</p> <p>False = Searching Library Hyperlinks is disabled, unless Hyperlinks is checked in the Advanced search.</p>	Boolean	"Using the Advanced Search Properties" on page 206 See Also: "EnableLibHyperlinksBtn" on page 191

Property	Description	Data Type	For more information, see
EnableLibHyperlinksBtn	Provides a check box so users can decide if they want to perform a Library Hyperlinks search. The default is True. True = The Hyperlinks check box is enabled in the advanced search properties. If used with EnableLibHyperlinks, the check box is checked by default. False = The Hyperlinks check box is disabled in the advanced search properties.	Boolean	"Using the Advanced Search Properties" on page 206
EnableLibImages	Enables users to search Library Images. The default is True. True = Searching Library Images is enabled. False = Searching Library Images is disabled, unless Images is checked in the Advanced search.	Boolean	"Using the Advanced Search Properties" on page 206 <i>See Also:</i> "EnableLibImageBtn" on page 191
EnableLibImageBtn	Provides a check box so users can decide if they want to perform a Library Images search. The default is True. True = The Images check box is enabled in the advanced search properties. If used with EnableLibImages, the check box is checked by default. False = The Images check box is disabled in the advanced search properties.	Boolean	"Using the Advanced Search Properties" on page 206
EnableLibraryFieldset	Creates a border around the group of library items in an advanced search. These items include: Images, Files and Hyperlinks. The default is False. True = Show border around library items. False = No border around library items.	Boolean	"Fieldset Properties" on page 212 <i>See Also:</i> "LibraryFieldsetLegend" on page 192

Property	Description	Data Type	For more information, see
FolderFieldsetLegend	Text that appears in the fieldset border around the Folder check boxes. This property is used only when the EnableFolderFieldset property is set to True. The default is: Folder Descriptions:	String	"Fieldset Properties" on page 212 See Also: "EnableFolderFieldset" on page 189
FolderID	The folder at which the search begins. The starting folder need not be the root folder. The Recursive property determines if the search examines this folder's subfolders.	String	"Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15 See Also: "Recursive" on page 194
Hide	Used to hide Search Server Control in design time and runtime. True = Hide control False = Show control	Boolean	
Language	Set a language for viewing search content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer	
LibraryFieldsetLegend	Text that appears in the fieldset border around the Library items check boxes. This property is used only when the EnableLibraryFieldset property is set to True. The default is: Library items:	String	"Fieldset Properties" on page 212 See Also: "EnableLibraryFieldset" on page 191

Property	Description	Data Type	For more information, see
LinkTarget	<p>Defines the way a link acts when a link is clicked. The choices are:</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>	String	
MaxCharacters	The maximum number of characters the input text box accepts when doing a search.	Integer	
MaxResults	<p>The maximum number of content blocks returned. (0=unlimited).</p> <p>When the EnableAjax property is set to true, the way this property works changes from the maximum amount of results to the maximum results per page. This takes advantage of Ajax's ability to have paging in the searches. If zero is entered, all results appear on the same page.</p>	Integer	"EnableAjax" on page 187
MaxTeaserLength	An optional search result formatting parameter that limits the length of any returned content item's teaser/summary. (0=unlimited)	Integer	

Property	Description	Data Type	For more information, see
OrderBy	<p>The order of the list to be returned. For example, you want to sort search results by last modified date.</p> <p>Title - The title of the content block</p> <p>ID - The XML content block ID number</p> <p>Date Created - The date the content block was created</p> <p>Date Modified - The date the content block was last modified</p> <p>LastEditorLName - The last editor's last name</p> <p>LastEditorFname - The last editor's first name</p> <p>Userld - The ID of the user who last edited the content</p>	String	
OrderDirection	<p>The direction the search results will be sorted in. The default is Ascending</p> <p>Ascending - Results from A to Z</p> <p>Descending - Results from Z to A</p>	String	
PostResultsAt	<p>Enter the Web form on which the search results will be posted.</p> <p>If this property is left blank, the results are posted to the Web form from which the search was initiated.</p> <hr/> <p>Important: A search control must exist on the posting page.</p> <hr/>	String	
Recursive	Whether to search sub-folders of the identified root folder. The root folder is identified in the <code>FolderID</code> property.	Boolean	See A/so: "FolderID" on page 192
RemoveTeaserHtml	<p>An optional search result formatting parameter that removes all the HTML from each content item's teaser/summary before sending it to the browser.</p> <p>True = Remove HTML</p> <p>False = Do not remove HTML</p>	Boolean	

Property	Description	Data Type	For more information, see
ResultTagId	Specify the div tag control ID to display the search results. This property is only active when the EnableAjax property is set to true.	String	"ResultTagId" on page 198 "EnableAjax" on page 187
ShowDate	If True , displays the content block's last modified date. This is useful for determining the most recent content block.	Boolean	"ShowDate Examples" on page 202
ShowExtendedSearch	<p>Allows a user to search via Searchable Property metadata.</p> <p>True = Show extended search False = Hide extended search</p> <p>Search type metadata only appears on the search screen if the following are true</p> <ul style="list-style-type: none"> • this property is set to True • the ShowSearchOptions property is set to True • at least one content item in the starting folder (and its subfolders if <code>recursive</code> is set to true) has searchable metadata applied to it <p>Note: The folderID property determines the search's starting folder.</p>	Boolean	"ShowExtendedSearch" on page 202 For more information on Metadata and Searchable Property, see the Administrator Manual section "Metadata".
ShowSearchBoxAlways	<p>If set to false, the search box does not appear on PostBack. The default is True.</p> <p>True = Display search box on PostBack False = Do not display search box on PostBack</p>	Boolean	

Property	Description	Data Type	For more information, see
ShowSearchOptions	<p>If set to true, the following choices appear along with the search box. This type of search is also known as a Standard search. The default setting is True.</p> <ul style="list-style-type: none"> • All the words • Any of the words • Exact phrase • Match Partial words <p>True = Show search options False = Hide search options</p>	Boolean	"Basic and Standard Searches" on page 208
Stylesheet	<p>Specify the location and style sheet for the submenu. For example: csslib\mytest.css. Leave blank to use the default. global.css is the default style sheet. It is located in \webroot\Workarea\csslib.</p>	String	"Using a Style Sheet" on page 216
SuppressWrapperTags	<p>Suppresses the output of the span/div tags around the control. The default is False.</p> <p>True - Suppress wrap tags. False - Allow wrap tags.</p>	Boolean	

Property	Description	Data Type	For more information, see
TargetFolderDescription List	<p>A comma delineated list of folders to search. The target folder list uses the Description property of a folder to define which folders are searched. This is an optional property. If you use this property, it will supersede the FolderID property. The default is blank. Below is an example of the syntax used.</p> <p>Football=1,News\Headlines=0</p> <p>Football, News and Headlines are the folder descriptions. The ones (1) and zeros (0) define whether a site visitor is forced to search the folder or not.</p> <p>1 = Automatically search this folder 0 = Do not automatically search this folder</p>	String	"TargetFolderDescription List" on page 205
Text Box Size	The size of the display text box for user input. The width is based on the number of characters.	Integer	"Text Box Examples" on page 199
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String	

Ajax Search

Ajax gives your site visitors an application-like search experience. For example, the search displays results with reloading the page. In addition, results can appear in a separate area of the Web form.

The Ajax search also lets you implement a paging feature. When `EnableAjax` is set to `true`, the `MaxResults` property determines the maximum number of results per page. See Also: "[MaxResults](#)" on page 193.

SEARCH

Search Text:

Total results:5
Viewing page 1 of 2
[\[First\]](#) [\[Previous\]](#) [\[Next\]](#) [\[Last\]](#)

Business Practices	ektron Medical's commitment to operating with integrity requires more than just adequate financial m...
Contact Form	
Ektron Rated Positive	Ektron Inc., an innovator in Web content management software, today announced ...

ResultTagId

By using the `ResultTagId` property, you can designate where you want search results to appear. This allows you to place the search criteria in one area of a Web form and the results in another. The `ResultTagId` property only works with the Ajax Search.

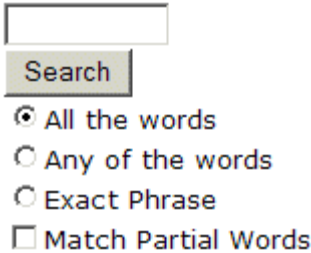
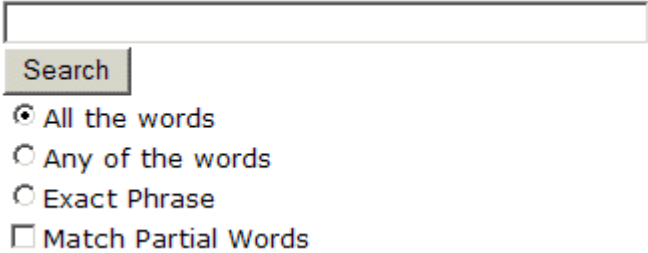
For example, you have the following Tag.

```
<span id="results"></span>
```

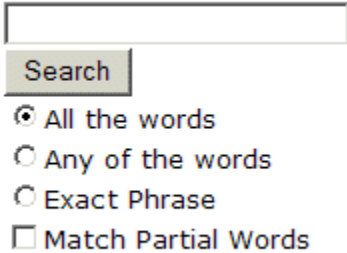
You would add **results** to the property, as shown below.

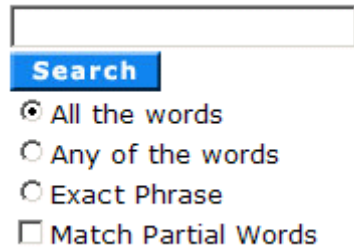
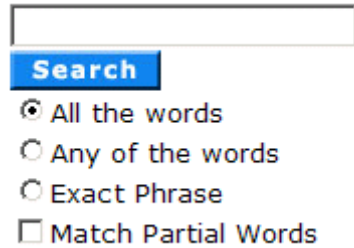
LinkTarget	_self
MaxCharacters	50
MaxResults	0
MaxTeaserLength	100
OrderBy	Title
OrderDirection	Ascending
PostResultsAt	
Recursive	True
RemoveTeaserHtml	True
ResultTagId	results
ShowDate	False
ShowExtendedSearch	True
ShowSearchBoxAlways	True
ShowSearchOptions	False

Text Box Examples

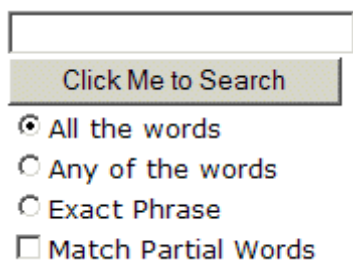
Value	Web Page View
10	
50	

Button Image Source Examples

Value	Web Page View
(blank)	

Value	Web Page View
search.jpg	 <p>A search interface with a text input field, a blue 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>
http://www.ektron.com/images/search.jpg	 <p>A search interface with a text input field, a blue 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>

Search Button Text Examples

Value	Web Page View
"Click Me to Search"	 <p>A search interface with a text input field, a button labeled 'Click Me to Search', and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>

Value	Web Page View
"Click Me to Search"	

Display Examples

Value	Web Page View
Vertical	
Horizontal	

ShowDate Examples

Value	Web Page View
False	<p>Search Results</p> <p> RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah </p>
True	<p>Search Results</p> <p> RC Sportster (8/27/2002 7:52:40 PM) RC Redstar (8/27/2002 7:52:01 PM) RC Lilly (8/27/2002 7:51:26 PM) RC Cheetah (8/27/2002 7:50:53 PM) Products Page (8/27/2002 7:50:09 PM) RC International Ships RC Sportster (8/27/2002 2:46:12 PM) RC International Hosts RC Conference (8/27/2002 2:44:44 PM) RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM) </p>

ShowExtendedSearch

ShowExtendedSearch lets you decide if the user can search the Searchable Property contained in the metadata of a content block. If you set the ShowExtendedSearch property to **true**, and a site visitor visits the search page, the extended search parameters appear below the search form.

Search

Text:





☒ All the words

☐ Any of the words

☐ Exact Phrase

☐ Match Partial Words [Advanced](#)

Audio Title:

Image Date: between [None]   and [None]  

Part Number: between and

The Searchable Properties for a folder must be turned on for them to appear in a search. For more information on using Searchable Properties, see the Administrator Manual section "Metadata."

The examples below show Searchable Properties being activated in the folder properties and appearing on a search page.

☐ Publish Office documents as PDF

Meta Data/Custom-Fields available for folder 'Content':

Assigned	Required	Name
<input checked="" type="checkbox"/>	<input type="checkbox"/>	title
<input checked="" type="checkbox"/>	<input type="checkbox"/>	keywords
<input checked="" type="checkbox"/>	<input type="checkbox"/>	description
<input checked="" type="checkbox"/>	<input type="checkbox"/>	page
<input checked="" type="checkbox"/>	<input type="checkbox"/>	robots
<input checked="" type="checkbox"/>	<input type="checkbox"/>	breadcrumb
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Audio Title
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Image Date
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Part Number

Notification Settings:

Address http://192.168.0.82/CMS400Example/webform5.aspx

Search

Text:

☒ All the words
☐ Any of the words
☐ Exact Phrase
☐ Match Partial Words [Advanced](#)

Audio Title:

Image Date: between [None] and [None]

Part Number: between and

TargetFolderDescriptionList

The TargetFolderDescriptionList property allows a developer to create a custom folder list for site visitors to search. Using this property is optional. If the property is used, it overrides the settings in the FolderID property.

☒ Football Stats Folders Only
☐ Home Folders Only

Search

Text:

☒ All the words

☐ Any of the words

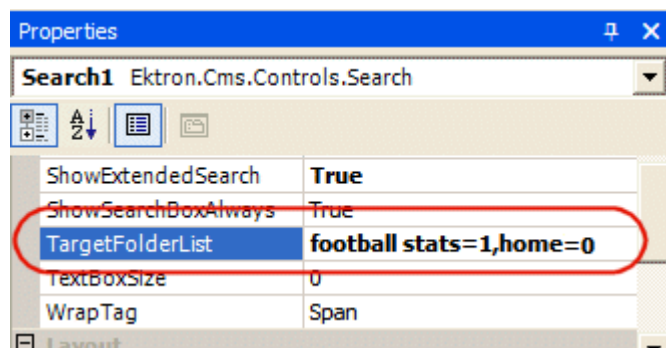
☐ Exact Phrase

☐ Match Partial Words

You can force site visitors to search only the specific folders you set in the TargetFolderDescriptionList. This is accomplished by creating a comma delineated folder list in the TargetFolderDescriptionList property and setting the EnableAdvancedLink property to False. When a visitor to your site performs a search, they only receive results from folders in the TargetFolderDescriptionList.

In contrast, you can give site visitors the ability to decide which folders they want to search. This is accomplished by creating a comma delimited folder list in the TargetFolderDescriptionList property and setting the EnableAdvancedLink property to True. This allows the site visitor to click the **Advanced** link and then choose from the folders you allow. The options appear as check boxes below the advanced properties

Here is an example of the TargetFolderDescriptionList in Visual Studio. If the Advanced link is disabled, a site user is forced to search only the Football Stats folder. If the Advanced link is enabled and a site visitor clicks it, he has the choice to search the Football Stats and Home. The **Football Stats** check box is checked by default, while the **Home** check box is not.



☒ Content ☒ Forms ☒ DMS Assets
☒ Images ☒ Files ☒ Hyperlinks
☒ Football Stats Folders Only
☐ Home Folders Only

Search

Text:

☒ All the words
☐ Any of the words
☐ Exact Phrase
☐ Match Partial Words

Using the Advanced Search Properties

There are three ways to use the advanced search properties. One way is to create a basic or standard search, in which the user must search by properties you choose. Another way is let users decide which properties they want use for the search. The last way is a mix of both. See Also: ["Basic and Standard Searches" on page 208](#), ["Advanced Search" on page 209](#) and ["Using Standard and Advanced Searches Together" on page 209](#)

The following is a list of advanced properties. These properties appear in the list of properties for the server control. For descriptions of these properties, see ["Search Server Control Properties" on page 183](#).

- EnableArchived
- EnableArchivedBtn
- EnableContent
- EnableContentBtn
- EnableDMSAssets
- EnableDMSAssetsBtn
- EnableForms
- EnableFormsBtn
- EnableLibFiles
- EnableLibFilesBtn
- EnableLibHyperlinks
- EnableLibHyperlinksBtn
- EnableLibImages

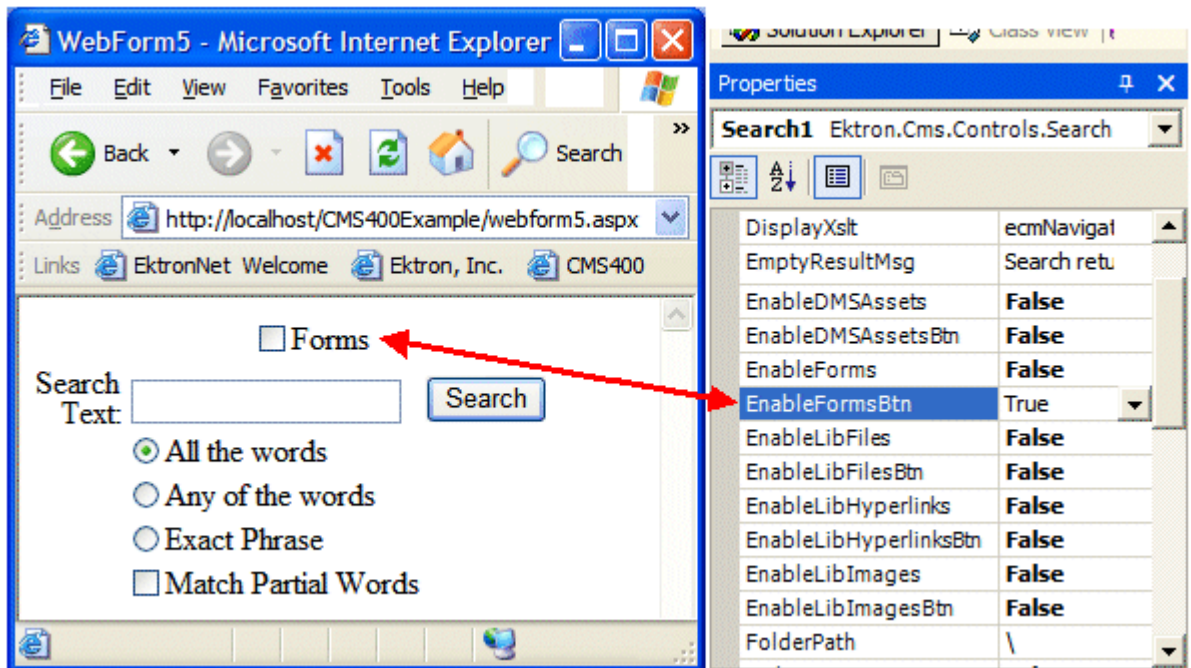
- EnableLibImagesBtn

WARNING! If you set all above properties to **false**, the search returns no results. This happens because there are no parameters to search against.

These properties can be used individually or in combination. For example, you could have a search that only searches Forms, or a search that searches Forms and DMS Assets.

Each advanced property has an **enable** check box property. These properties have **Btn** at the end of their name. They create a check box that appears when the Advanced link on a search is clicked. The check boxes only appear when the properties are set to true. Using the Advanced link and check boxes to search is known as an Advanced Search. See Also: ["Advanced Search" on page 209](#)

The following example shows the Forms check box enabled on the site search and the EnableFormsBtn property set to True in Visual Studio.



You can force a user to search using only properties you define, preventing him from changing any properties. For example, to create a forms-only search, set all properties with **Enable** in the name to **false** except for EnableForms. This is known as a standard search. See Also: ["Basic and Standard Searches" on page 208](#)

NOTE When setting up a server control for a standard search, you must choose the actual property to search with, not the property that enables the check box. For example, to search forms, use the EnableForms property, not the EnableFormsBtn property.

Basic and Standard Searches

A basic search contains only three items when displayed on a page:

- text to describe the text box
- a search text box
- a submit or “search” button

A basic search takes the least amount of real estate on your page and provides the fewest number of options to a site visitor. Below is an example of a basic search. This is handy when you want to have a search perform specified functions with little user control. To make your search control appear as a basic search, set the ShowSearchOptions property to **false**.



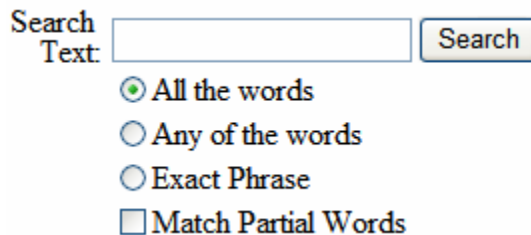
A basic search control consisting of the text "Search Text:" followed by a text input box and a "Search" button.

A standard search allows a site user a few more options when searching your site. These options, listed below, allow a site user to decide how the text should match the information they are searching for.

- All the words
- Any of the words
- Exact phrase
- Match Partial words

With basic and standard searches, you decide which properties are used in the search. At least one advanced search property must be set to **true** in the server control. Otherwise, the search returns no matches.

When added to a template and visited, a standard search looks like this.



A standard search control. It includes the text "Search Text:" followed by a text input box and a "Search" button. Below the input box are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".

Users enter text, change the way the text matches the items being searched, and click the search button to search for the types of content you allow. For examples of setting up a standard search, see ["Using Standard and Advanced Searches Together"](#) on page 209

Advanced Search

In an advanced search, you decide which content types are available. From those choices, site visitors decide which content types they want and perform a search.

To let a site visitor choose which content types to search, the Advanced Hyperlink must be turned on. Otherwise, users can only search properties you have enabled. See [A/so: "EnableAdvancedLink" on page 187](#)

When the Advanced hyperlink is enabled, this is how the search appears.

Search Text:

☒ All the words
☐ Any of the words
☐ Exact Phrase
☐ Match Partial Words

[Advanced](#)

When the Advanced hyperlink is clicked, this is how the search appears. This assumes all choices are enabled and automatically checked. For examples of setting up an advanced search, see ["Using Standard and Advanced Searches Together" on page 209](#)

☒ Content ☒ Forms ☒ DMS Assets
☐ Images ☐ Files ☐ Hyperlinks
☐ Intranet Folders Only



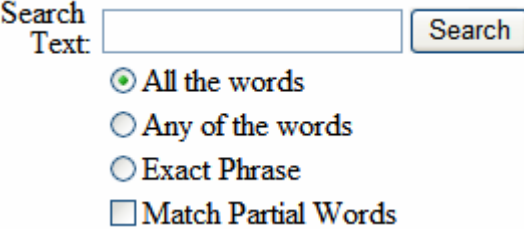
Search Text:

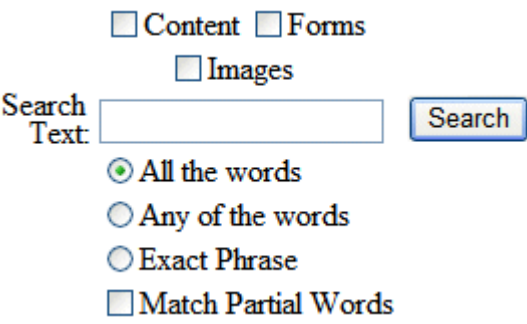
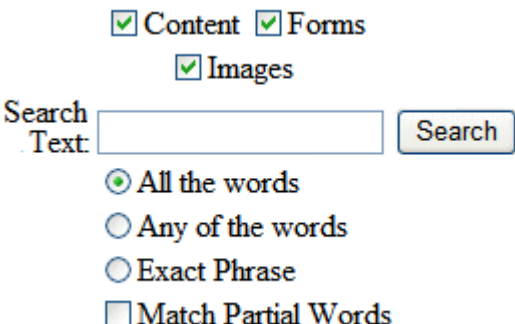
☒ All the words
☐ Any of the words
☐ Exact Phrase
☐ Match Partial Words

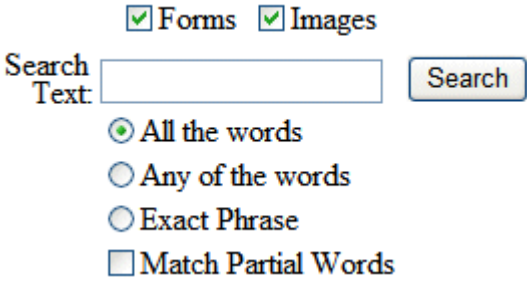
Using Standard and Advanced Searches Together

You can use a mix of the standard search and advanced search to force people to search by certain properties while giving them a choice on others. For example, you want users to always search Content, yet give them an option to search forms, files, and images.

The following examples illustrate how to configure the Search server control using the advanced properties. If a property is not mentioned in the examples below, use its default setting.

Description	How the Search Appears to Site Visitors	Property Settings
<p>Search content only. Do not let the site user change advanced settings. No standard options appear.</p> <p>This is a Basic search. See Also: "Basic and Standard Searches" on page 208</p>		<p>EnableAdvancedLink = False EnableContent = True Set all other properties with the word Enable to False.</p>
<p>Search content, forms, and library images only. Do not let the user change advanced settings. No standard options appear.</p> <p>This is a Basic search. See Also: "Basic and Standard Searches" on page 208</p>		<p>EnableAdvancedLink = False EnableContent = True EnableForms = True EnableLibImages = True Set all other properties with the word Enable to False.</p>
<p>Search content only. Do not let the site user change advanced settings. Standard options appear.</p> <p>This is a Standard search. See Also: "Basic and Standard Searches" on page 208</p>		<p>EnableAdvancedLink = False EnableContent = True ShowSearchOptions = True Set all other properties with the word Enable to False.</p>

Description	How the Search Appears to Site Visitors	Property Settings
<p>Let the user choose whether to search content, forms, library images, or all of them together. The user can also mix and match properties. Standard options appear.</p> <p>For example, search content and library images, but not forms.</p> <p>This is an Advanced search. See Also: "Advanced Search" on page 209</p> <hr/> <p>Warning: If a user does not check at least one box, the search returns no results.</p> <hr/>	 <p>The screenshot shows a search interface with three unchecked checkboxes: <input type="checkbox"/> Content, <input type="checkbox"/> Forms, and <input type="checkbox"/> Images. Below these is a search text input field and a 'Search' button. Underneath the input field are four radio button options: <input checked="" type="radio"/> All the words, <input type="radio"/> Any of the words, <input type="radio"/> Exact Phrase, and <input type="checkbox"/> Match Partial Words.</p>	<p>EnableAdvancedLink = True EnableContentBtn = True EnableFormsBtn = True EnableLibImagesBtn = True ShowSearchOptions = True Set all other properties with the word Enable to False.</p>
<p>The check boxes are checked by default. The user can change the check boxes. Standard options appear.</p> <p>This is an Advanced search. See Also: "Advanced Search" on page 209</p> <hr/> <p>Warning: If a user unchecks all of the boxes, the search returns no results.</p> <hr/>	 <p>The screenshot shows a search interface with three checked checkboxes: <input checked="" type="checkbox"/> Content, <input checked="" type="checkbox"/> Forms, and <input checked="" type="checkbox"/> Images. Below these is a search text input field and a 'Search' button. Underneath the input field are four radio button options: <input checked="" type="radio"/> All the words, <input type="radio"/> Any of the words, <input type="radio"/> Exact Phrase, and <input type="checkbox"/> Match Partial Words.</p>	<p>EnableAdvancedLink = True EnableContent = True EnableContentBtn = True EnableForms = True EnableFormsBtn = True EnableLibImages = True EnableLibImagesBtn = True ShowSearchOptions = True Set all other properties with the word Enable to False.</p>

Description	How the Search Appears to Site Visitors	Property Settings
<p>The user is forced to search Content but can choose to search Forms or Images. The Forms and Images check boxes are checked by default. Standard options appear.</p> <p>This is a mix of Standard and Advanced searches. See Also: "Using Standard and Advanced Searches Together" on page 209</p> <hr/> <p>Warning: If a user unchecks all of the boxes, the search still search content and will return any matching results.</p> <hr/>		<pre> EnableAdvancedLink = True EnableContent = True EnableForms = True EnableFormsBtn = True EnableLibImages = True EnableLibImagesBtn = True ShowSearchOptions = True Set all other properties with the word Enable to False. </pre>

Fieldset Properties

NOTE

The properties described below can be seen by a site visitor only when the Advanced link is enabled and clicked.

The fieldset properties let you refine the look and feel of your site search by separating items used in the Advanced search into groups. Each group can have its own legend. The legends make it easier for site visitors to understand their choices and how those choices affect the search.

The fieldset properties are:

- ContentFieldsetLegend
- EnableContentFieldset
- FolderFieldsetLegend
- EnableFolderFieldset
- LibraryFieldsetLegend
- EnableLibraryFieldset

These properties can be broken down into two types:

- enabling fieldsets
- enabling fieldset legends

When you enable a fieldset, a border is created around a group of items. For example, if the `EnableContentFieldset` property is set to **true**, a border surrounds the different types of content.

The screenshot shows a search control interface. At the top, there is a yellow-bordered box containing six items, each with a green checkmark icon: **Content**, **Forms**, **DMS Assets**, **Images**, **Files**, and **Hyperlinks**. Below this box is a "Search Text:" label followed by a text input field and a "Search" button. Under the input field are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".

You can add a legend to the fieldset to describe its content. For example, by adding **Content Types:** to the `ContentFieldsetLegend` property, you create a legend for the fieldset around the different types of content.

The screenshot shows a search control interface with a legend. At the top, there is a yellow-bordered box. Inside the box, the text "Content Types:" is written in blue. Below it are six items, each with a green checkmark icon: **Content**, **Forms**, **DMS Assets**, **Images**, **Files**, and **Hyperlinks**. Below the box is a "Search Text:" label followed by a text input field and a "Search" button. Under the input field are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".

Retrieving the XML Structure of a Search

Retrieving the XML structure of XML content provides greater control over developing XSLs. The following is an example of how to retrieve the XML structure:

1. Open a new Web form.
2. Drag and drop a Search server control onto it.
3. Set the FolderID property.
4. Drag and drop a Textbox on the Web form.
5. Set the TextMode property to MultiLine.

NOTE It is also recommended that you set the width of the text box to at least 400px.

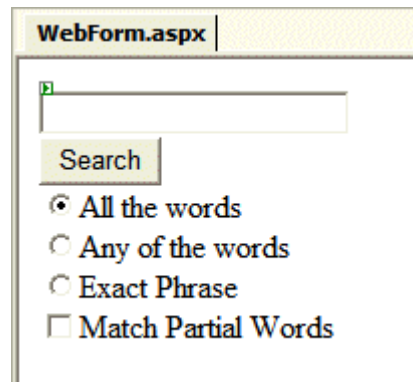
6. On the code behind page, add the following line.

```
Textbox1.Text = Search1.XmlDoc.InnerXml
```
7. Build the project.
8. View the Web form in a browser.
9. Search for something.
10. When the search returns any items, the XML structure of those items appear in the textbox.

Displaying Search Results on a Separate Page

The following example shows a search dialog box on a Web form, and the results displayed on separate Web form. This is done when you the search dialog box appears in a navigation pane, and the results appear on a separate Web form.

1. Drag and drop a Search server control on a Web form.

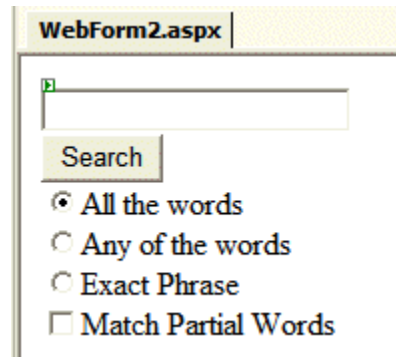


2. Set the following Ektron properties for the Search server control.
EnableAjax = False
DisplayXslt = None - Databind Only
PostResultAt = The Web form that displays the search results. For example, `Webform2.aspx`

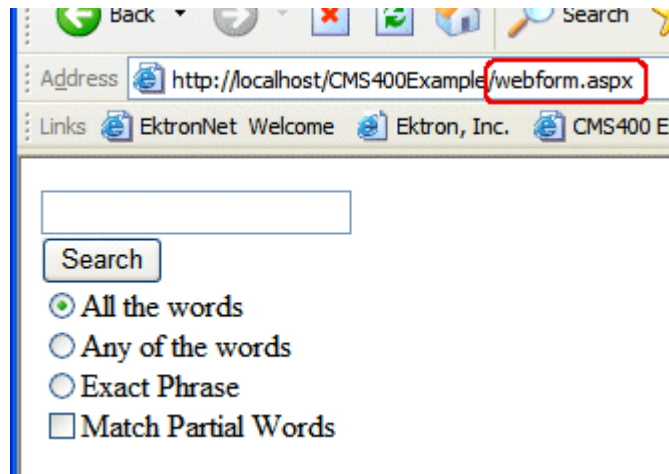
NOTE `webform2.aspx` represents the name of the second Web form.

3. Add a second Web form to the project.

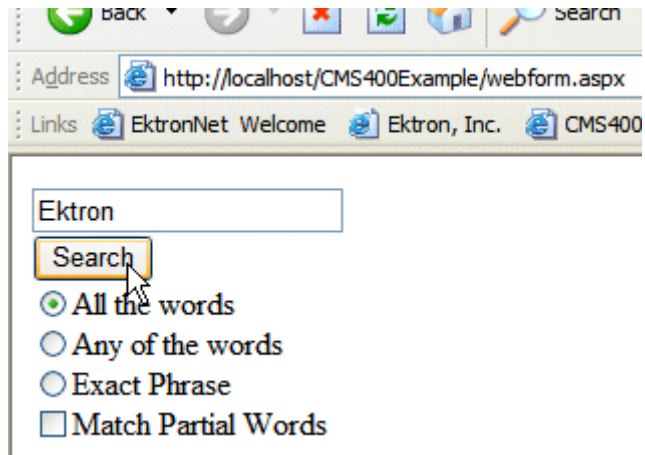
4. Drag and drop a Search server control on it.



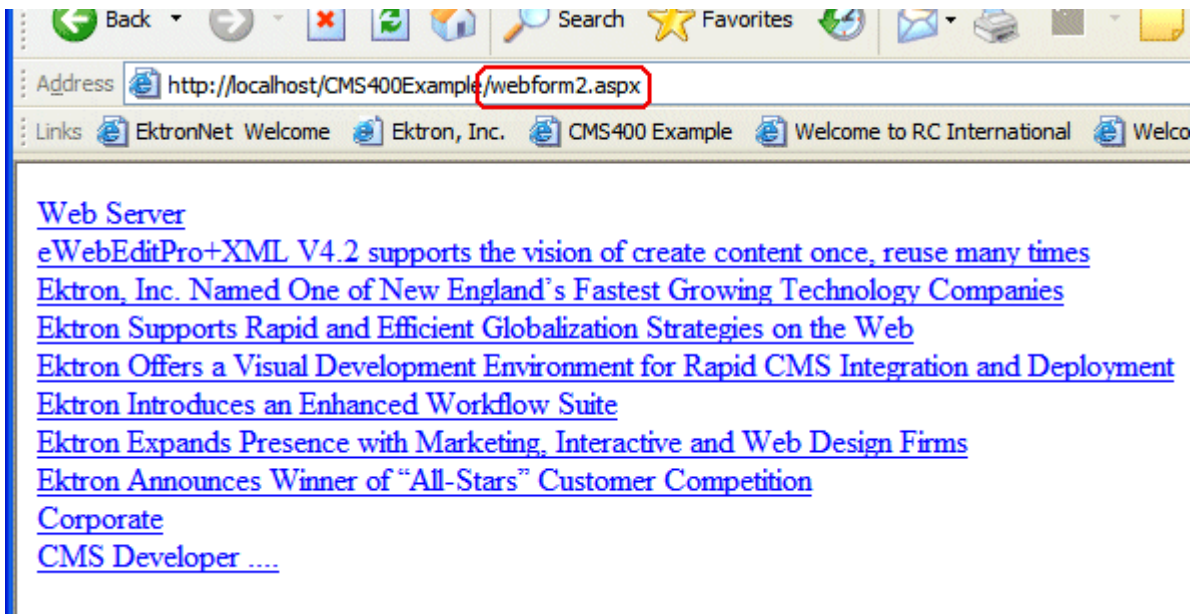
5. Set the Ektron property **ShowSearchBoxAlways** to **false**.
6. Build the solution.
7. In your browser, go to the first Web form.



8. Use the search dialog.



9. The search results display on the second Web form.



Using a Style Sheet

You can create your own style sheet, or modify the existing default one, `global.css`. It is located in `\webroot\cms400demo\Workarea\csslib`.

Style Sheets let you choose different text colors, text alignment, text fonts, and text size to create a customized feel for your search.

Best Practice

Ektron recommends copying and renaming the default style sheet. This insures you always have a clean file to start with. This also gives you something to reference if you are not getting the results you expect.

The style sheet must be located in the site root.

You can only define one Search style sheet for each Web page. If you have more than one Search on a single Web page, they share the style sheet.

For more information on style sheets, visit <http://www.w3.org/Style/CSS/>.

Formatting the Search Server Control

Additional information on formatting the Search server control can be found in the following Knowledge Base article.

<http://www.ektron.com/developers/cms400kb.cfm?id=2322>

IndexSearch Server Control

Use the IndexSearch server control to display an Smart Form's search screen on any Web form. This lets a user search your site based on XML parameters.

See *Also*: The **Ektron CMS400.NET** Administrator Manual section "XML Indexing".

Below is a sample screen that can be generated via the IndexSearch server control.

The form contains the following elements:

- Genre:** A dropdown menu with 'novel' selected.
- Price:** A dropdown menu with 'Between' selected, followed by a text input containing '1' and an empty text input.
- Last Name:** A dropdown menu with 'Jones' selected.
- Paperback:** A checkbox that is currently unchecked.
- Publishdate:** A dropdown menu with 'Select' selected, followed by a text input and a small calendar icon.
- Title:** A dropdown menu with 'Contains' selected, followed by a text input containing 'cms'.
- Submit Search:** A button located at the bottom center of the form.

You can also predefine a search and have the results appear on a Web form template by using the SearchParmXML property.

This subsection contains the following topics:

- ["IndexSearch Server Control vs. Search Server Control" on page 218](#)
- ["IndexSearch Server Control Properties" on page 218](#)
- ["SearchParmXML Property" on page 225](#)
- ["Using the IndexSearch Server Control Programmatically Example" on page 228](#)
- ["Programmatically Predefined General Search Results Replaced By Specific Search Results" on page 231](#)
- ["Turn Off Caching In Netscape Navigator and Mozilla Firefox" on page 234](#)

IndexSearch Server Control vs. Search Server Control

See, ["Search Server Control vs. IndexSearch Server Control" on page 183.](#)

IndexSearch Server Control Properties

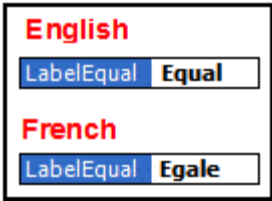
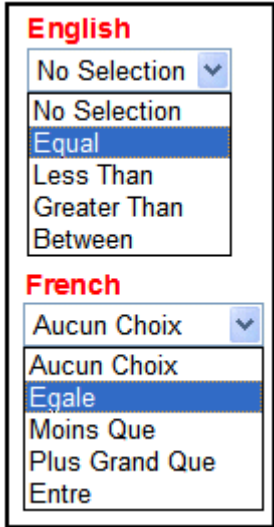
The IndexSearch server control properties are described in this table.

Note The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

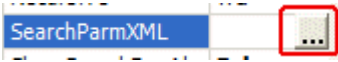
Property	Description	Data Type	Values
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderID property. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String	
ButtonImgSrc	If you want to display an image on the submit button, enter the server path to it. See Also: "Button Image Source Examples" on page 199 .	String	
ButtonText	The text that appears on the button.	String	The default is: Search
ContentParameter	Checks the QueryString for the content block ID value and replaces the search with content when ID is specified. Leave blank to always display the search.	String	None - Use Default ID - The server control uses the content block passed as a URL parameter.

Property	Description	Data Type	Values
DisplayXslt	Determines how the information on the page is displayed	String	<p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder</p> <p>See Also: "Example of ecmNavigation Display" on page 85</p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary</p> <p>See Also: "Example of ecmTeaser Display" on page 85</p> <p>Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page</p>
EmptyResultMsg	The message that displays if the search returns no hits.	String	The default value is: Your search did not match any documents.
EncodeContentHtml	When set to True, Content HTML is encoded when it is returned.	Boolean	<p>True = Content HTML is encoded</p> <p>False = Content HTML is not encoded</p>
FolderId	<p>The numeric id of the folder that you want to search. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15</p> <p>The Recursive property determines whether the IndexSearch server control will use this folder's child folders in the search.</p>	Integer	<p>Any folder's numeric ID</p> <p>0 (zero) represents the root folder.</p>
Hide	Used to hide output of the index search in design time and run time. For example, you want to pass the results to code behind for further manipulation.	Boolean	<p>True = Hide results</p> <p>False = show results</p>

Property	Description	Data Type	Values
HitsMarker	<p>This symbol shows how many search criteria each hit matched.</p> <p>The following example shows the first hit met three of the criteria. The second group met one criterion.</p> <div> <div>***</div> <div> Microsoft ASP.NET Programming Content Management Bible Visual Basic.NET How to Program ASP.NET Unleashed Simply Visual Basic.NET 2nd Edition </div> </div> <p>This property is used when the weighted property is set to true. See Also: "Weighted" on page 225.</p>	String	<p>* # +</p> <p>(Custom) Custom lets you choose the marker to use. The custom marker can be any text, numeric, or symbol character except & and <.</p> <p>For more Information, see The Ektron CMS400.NET Administrator Manual section "XML Indexing" > "Displaying Search Results"</p>

Property	Description	Data Type	Values
Labelafter Labelbefore LabelBetween LabelContains LabeldateBetween LabelEqual LabelExactPhrase LabelGreaterThan Labelinputdate Labelinputnumber LabelLessThan Labelon LabelSelect	<p>All of these properties let you change their label names. This is useful for Multi-language issues. For example, if you have a French site, you change Equal to Égale.</p>  <p>Note: These label properties will not change based on the Language property's setting. For example, setting the Language property to 1036 will not change these labels to French.</p>	String	<p>Examples of what the user will see on the site in French and English.</p> 
Language	Set a language for viewing index search results. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer	

Property	Description	Data Type	Values
LinkTarget	Defines the way a link acts when a link is clicked.	String	<p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>
MaxResults	The Maximum number of content blocks returned (0=unlimited).	Integer	

Property	Description	Data Type	Values
OrderBy	The order of the list to be returned. For example, you want to sort search results by last modified date.	String	<ul style="list-style-type: none"> • Title - The title of the content block • ID - The XML content block ID number • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • AuthorLName - The last author's last name • AuthorFname - The last author's first name • StartDate - The GoLive Date
OrderByDirection	The direction the results will be sorted in. This property works with the OrderBy property only.	String	Ascending Descending
Recursive	Determines whether the IndexSearch server control will use this folder's child folders in the search.	Boolean	True False
SearchParmXML	<p>Opens a separate window that lets you predefine a search. For example, you want a predefined list of books to appear on a web page.</p> <p>To access the window click the gray button.</p> 		For more information on SearchParmXML, see "SearchParmXML Property" on page 225
ShowSearchBoxAlways	If set to false , the search box does not appear onPostBack	Boolean	True False

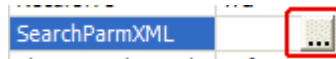
Property	Description	Data Type	Values
Weighted	<p>If you set this value to true, content blocks are ranked by the number of criteria matches.</p> <p>For example, if a content block matches all search criteria, it appears at the top of the results page. A hit marker appears for each match.</p> <p>If a second content block only matches one criterion, it also appears on the results page underneath the first content block. One asterisk appears above the page's title to indicate the single match.</p>	Boolean	<p>True - display every content block that matches <i>at least one</i> search criteria</p> <p>False - show only content blocks that match <i>all</i> search criteria</p>
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p>	String	<p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>
xmlConfigID	ID of the Smart Form whose indexed fields will appear on the search screen.	Integer	

SearchParmXML Property

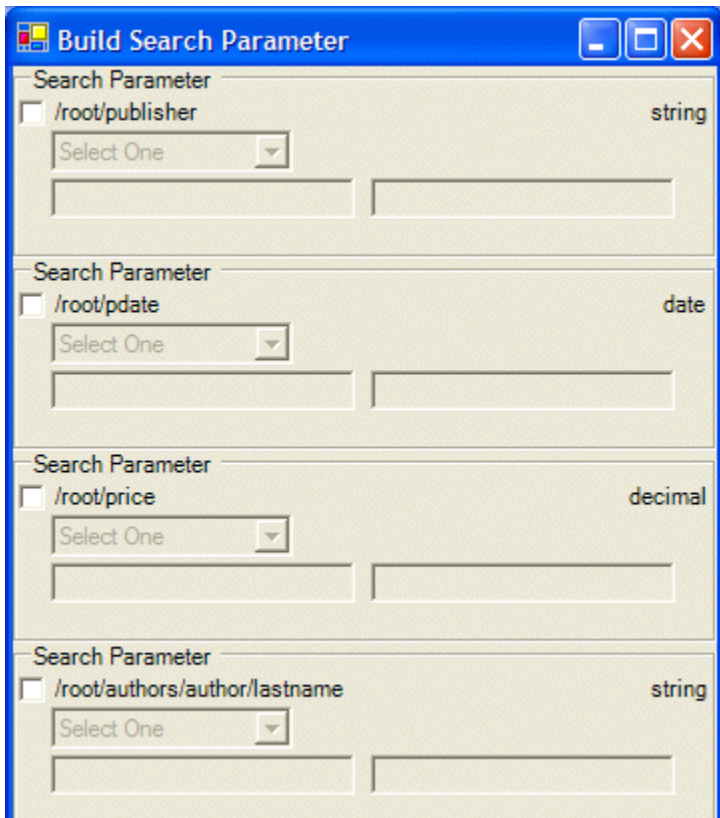
The SearchParmXML property is used to create a predefined XML search, whose results appear upon page load. Clicking on the gray box in this property opens a window, where you define the search. The parameters in this window are defined by the xmlConfigID you choose. Using this property launches the CMS Explorer login, if you are not already logged in. For more information on the CMS Explorer login, see ["Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15](#)

The following example uses the SearchParmXML property to create a list of books, based on publishers with "press" in their name.

1. To activate the window, click the gray box in the SearchParmXML property.



2. The Build Search Parameter screen appears.



Build Search Parameter

☐ Search Parameter /root/publisher string

Select One

☐ Search Parameter /root/pdate date

Select One

☐ Search Parameter /root/price decimal

Select One

☐ Search Parameter /root/authors/author/lastname string

Select One

- Click the check boxes for the parameters you want to search with.

The screenshot shows the 'Build Search Parameter' dialog box with four sections, each for a different search parameter. The first section, for '/root/publisher' (string type), has its checkbox checked and is highlighted with a red box. The other three sections are for '/root/pdate' (date), '/root/price' (decimal), and '/root/authors/author/lastname' (string), all with unchecked checkboxes. Each section contains a 'Select One' dropdown menu and two empty text input fields.

- Define the parameter.

This screenshot shows the same dialog box, but now the dropdown menu for the selected '/root/publisher' parameter is set to 'Contains' and the text input field contains the word 'Press'.

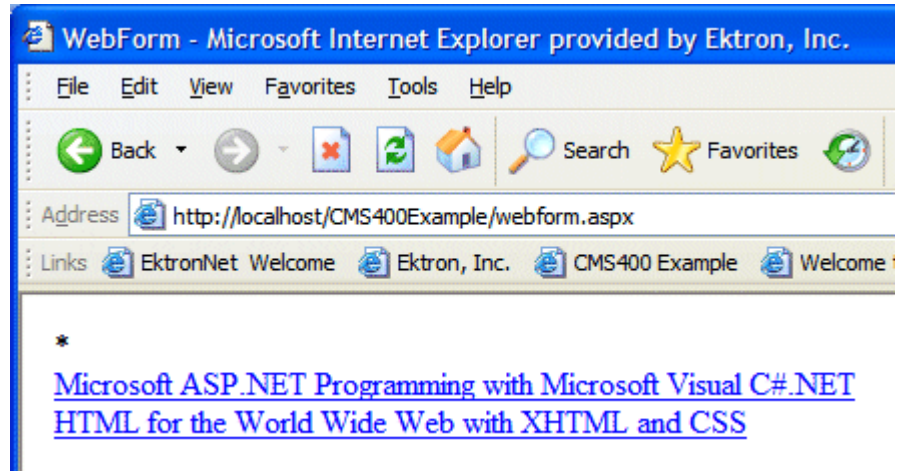
The drop down box in this example contains the following variables:

Select One - Prompts you to choose a variable.

Exact Phrase - You enter a phrase and search for that exact phrase, contained within the XML content.

Contains - You enter a word or phrase and search for any XML content, that has that word or phrase contained within it.

- Click OK.
- Build the Web form.
- Browse to the Web page in your browser.



8. The page appears with a list of XML content blocks that match the parameters you set.

Using the IndexSearch Server Control Programmatically Example

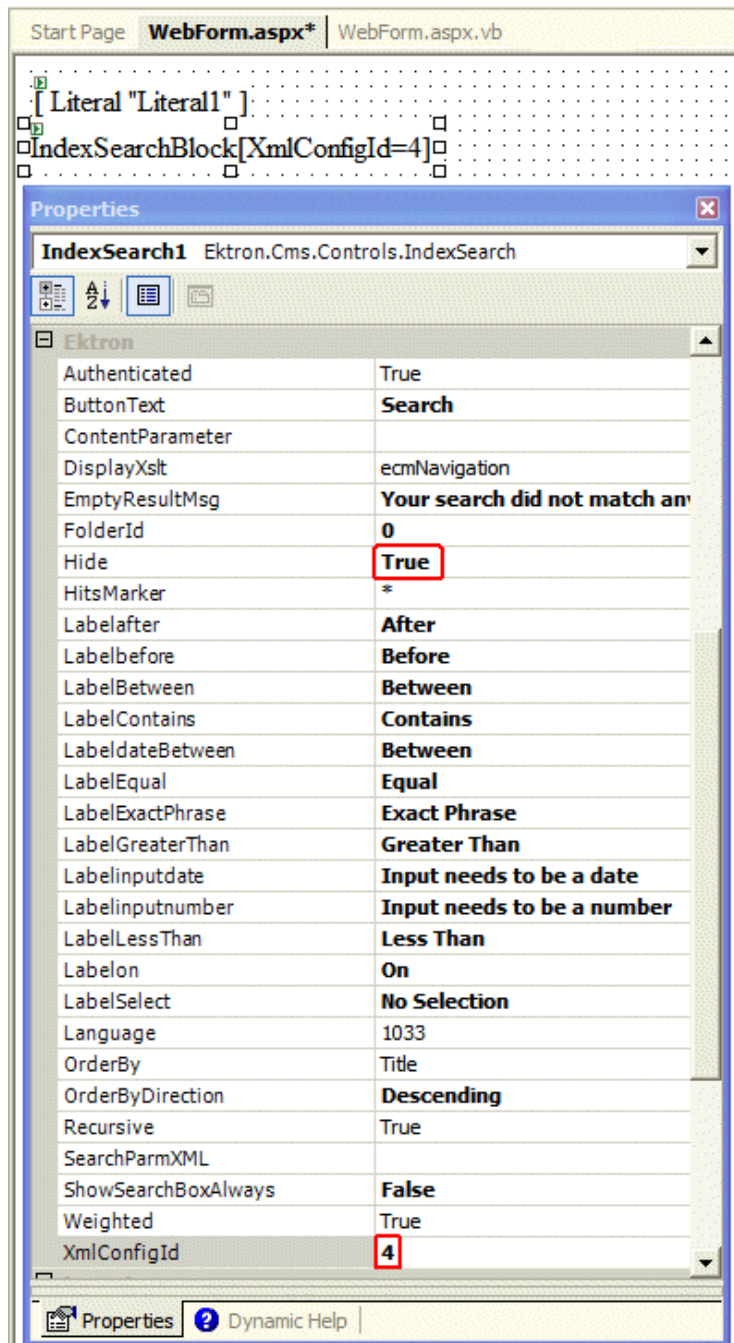
NOTE When creating an IndexSearch server control form user interface using code behind only, you need the IndexSearch server control to be connected with the page events. Otherwise you will not be able to submit your search parameters.

You can use the IndexSearch server control, in code behind, to manipulate the way users view the search results. The following example, using code behind, shows how to return values associated with an XPath. This example shows a list of book subject titles.

1. Open a Web form.
2. Drag and drop an IndexSearch server control on your Web form.
3. Set the properties in the properties window.

NOTE In this example, the hide setting must be set to **true**. Remember to set the XmlConfigID. This example uses ID number **4**.

4. Drag and drop a Literal on the Web form.



5. Add the following code to the code behind.

```
IndexSearch1.Parm.XPath = "/root/subject"
Dim arStr As String()
Dim Str As String
```

```

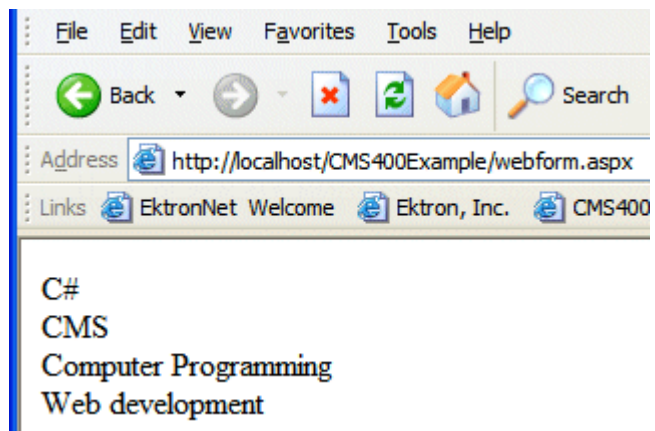
Dim strRet As String
arStr = IndexSearch1.GetXPathValues("/root/subject")
For Each Str In arStr
    strRet += Str & "<br/>"
Next
Literal1.Text = strRet

```

Here is a breakdown of the code.

Code snippet	Description
IndexSearch1.Parm.XPath = "/root/subject"	Defines the parameter XPath location:
Dim arStr As String() Dim Str As String Dim strRet As String	Creates variables
arStr = IndexSearch1.GetXPathValues("/root/subject")	Sets the variable arStr as an array of the XPathValues
For Each Str In arStr strRet += Str & " " Next	The for next loop iterates through the array of XPathValues
Literal1.Text = strRet	Displays values on your Web form

6. Build the project.
7. Browse to your web form.
8. The list of book subject titles appears.



Programmatically Predefined General Search Results Replaced By Specific Search Results

This Code behind example shows a predefined search appearing on a Web page, then being replaced by more specific search results. This example creates a hyperlink list of book publishers. When you click a specific publisher, it replaces the publisher list with a book list from that publisher.

Follow these steps.

1. Add an IndexSearch server control to a Web form.
2. Set the following properties:
 - DisplayXslt = ecmTeaser
 - Hide = True
 - HitsMarker = (blank)
 - ShowSearchBoxAlways = False
 - Weighted = True
 - XmlConfigId = 4
3. Add two Literals.
4. Add the following code to the Page Load Event in code behind.

```
Dim bRet As Boolean
    If Request.QueryString("publisher") <> "" Then
        IndexSearch1.Hide = False
        IndexSearch1.Parm.XPath = "/root/publisher"
        IndexSearch1.Parm.DataType =
Ektron.Cms.Common.EkEnumeration.XMLDataType.String
        IndexSearch1.Parm.SearchType =
Ektron.Cms.Common.EkEnumeration.XMLSearchRangeType.ExactPhrase
        IndexSearch1.Parm.Value1 = Request.QueryString("publisher")
        bRet = IndexSearch1.AddParm()
        IndexSearch1.Search()

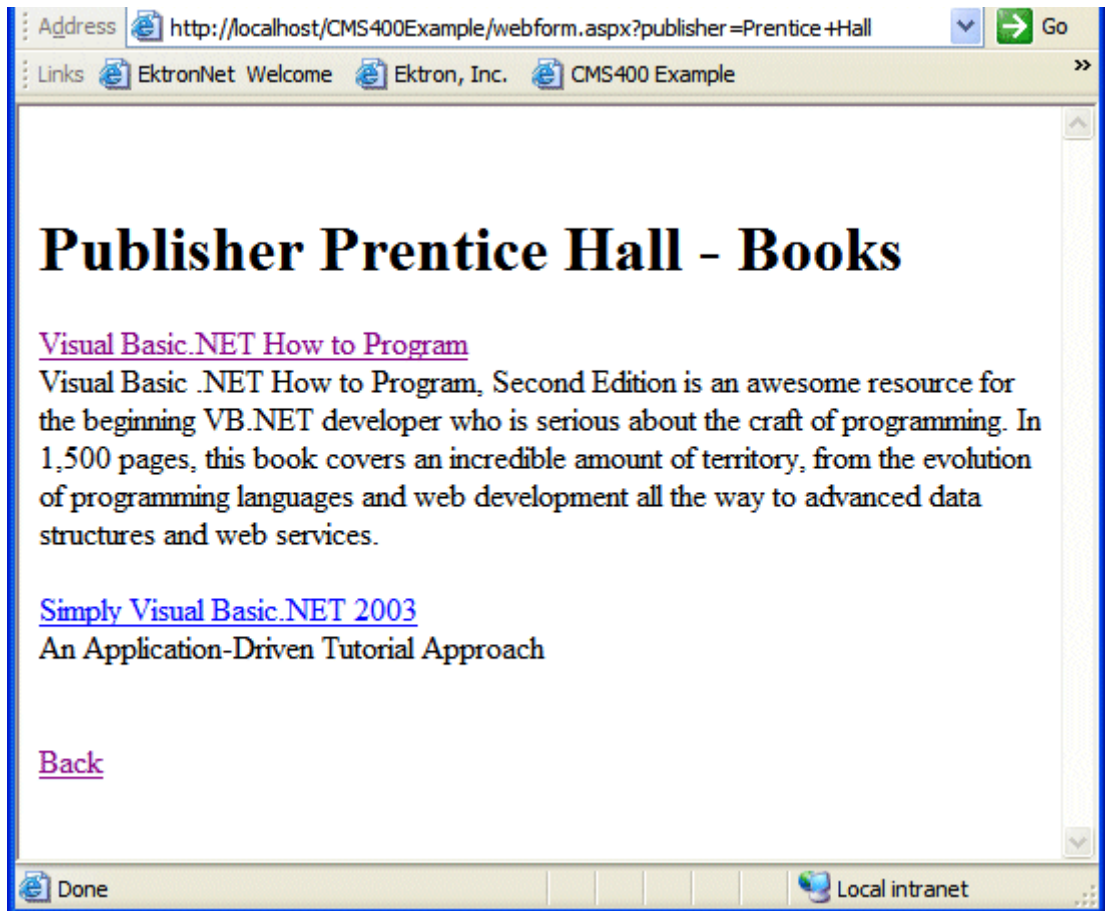
        Literal1.Text = "<h1>Publisher " & Request.QueryString("publisher") & " -
Books</h1>"
        Literal2.Text = "<a href=\""webform.aspx\"">Back</a>"
    Else
        Literal1.Text = "<h1>Publishers</h1>"
        Dim arStr As String()
        Dim Str As String
        Dim strRet As String
        arStr = IndexSearch1.GetXPathValues("/root/publisher")
        For Each Str In arStr
            strRet += "<a href=\""webform.aspx?publisher=\"" & Server.UrlEncode(Str) &
"">" & Str & "</a><br/>"
        Next
        Literal1.Text += strRet
    End If
```

5. Save your Web form.
6. Build your Web form.
7. Browse to the Web form in your browser.
8. The list of publishers appears.



9. Click on a publisher.

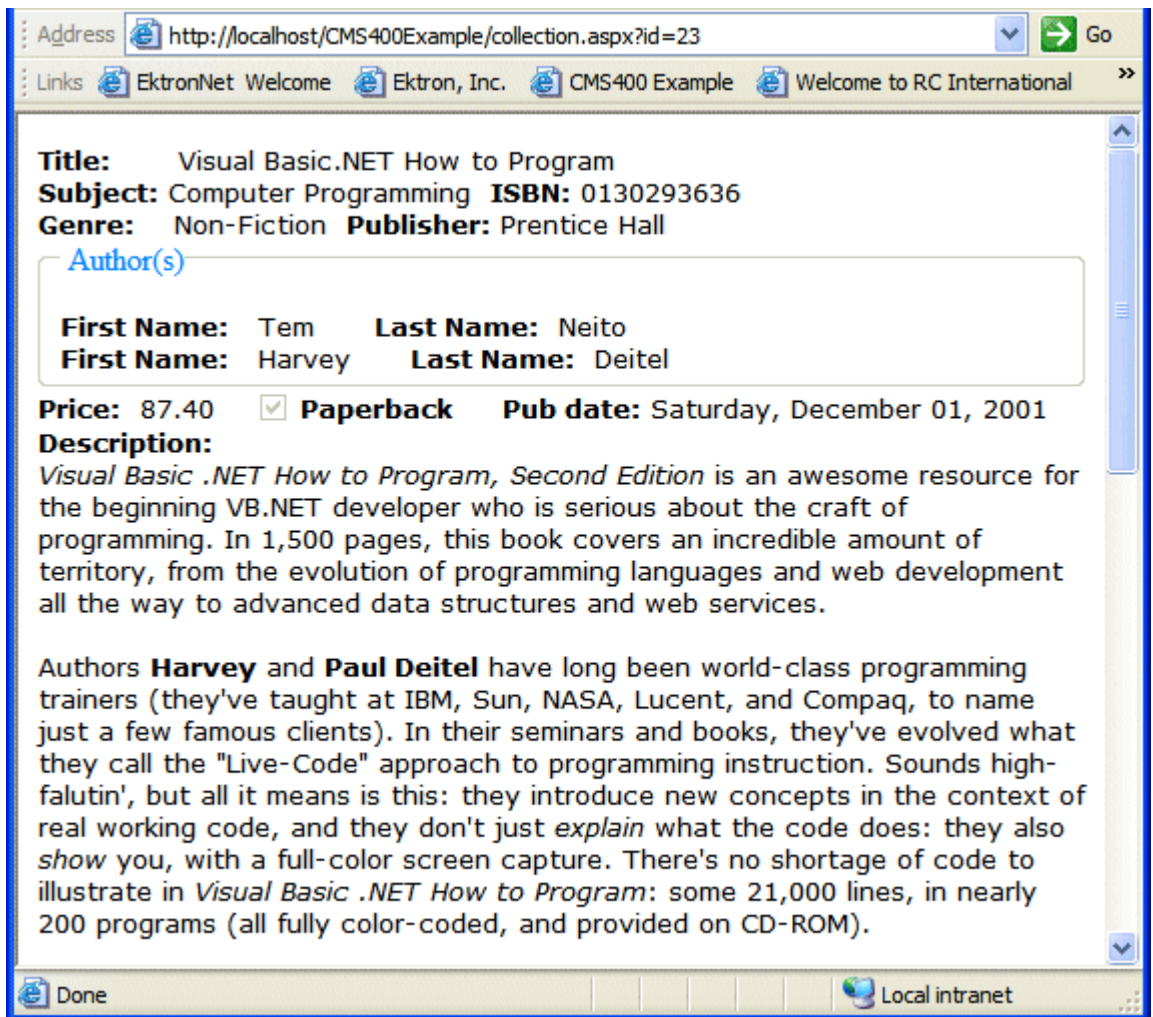
10. A list of books with teasers by that publisher appears.



Notice the second Literal now contains a hyperlink to go back.

11. Click on a book.

12. The XML Content Block for that book appears.



Turn Off Caching In Netscape Navigator and Mozilla Firefox

When using Netscape Navigator or Mozilla Firefox to perform an index search some parameters become cached. To prevent caching in Netscape Navigator and Mozilla Firefox, add the following line to the Page Load Event in code behind:

```
Response.Cache.SetNoStore( )
```

SiteMap Server Control

The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap. The sitemap is displayed as indented list when viewed on a Web page.



The contents of the sitemap are defined on the Breadcrumb tab located in the Workarea folder properties. See Also: The Administrator Manual > “Managing Content Folders” > “Breadcrumbs”.

This subsection contains the following topics:

- ["Sitemap Server Control Properties" on page 235](#)
- ["Using the Sitemap Server Control" on page 237](#)

Sitemap Server Control Properties

The Sitemap server control properties are described in this table.

Note

The following table only lists Ektron-specific properties. It does not describe native .NET properties such as font, height, width and border style. For documentation of these properties, see VisualStudio help.

Property	Response	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 15	String
CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See <i>Also</i> : "Caching with Server Controls" on page 36	Integer
ClassName	The style sheet class name used to format the HTML. Leave blank to use the default. To use a new class, add it to the following file: <code>webroot\CMS400Demo\Workarea\csslib\sitemap.css</code> Then, add the class name to the property.	String
FolderID	The folder ID for the starting point of the sitemap. To choose the root folder, enter 0 (zero).	Integer
Hide	Used to hide the Sitemap in design time and run time. True = Hide Sitemap trail False = Show Sitemap trail	Boolean
Language	Set a language for viewing content. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).	Integer
MaxLevel	Set the maximum amount of levels to show. Set it to 0 (zero) for unlimited.	integer
SuppressWrapperTags	Suppresses the output of the span/div tags around the control. The default is False . True - Suppress wrap tags. False - Allow wrap tags.	Boolean

Property	Response	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Using the Sitemap Server Control

NOTE Make sure you have added the sitemap information to your folders' Breadcrumb tab in the CMS400.NET Workarea.

Follow these steps to use the Site server control.

1. Open a Web form for which you want to create a sitemap.
2. Drag and drop the Sitemap server control onto an appropriate location of the Web form.
3. Add the starting folder's ID to the **FolderID** property.
4. Set any of the other properties. See Also: "[Sitemap Server Control Properties](#)" on page 235.
5. Save the Web form.
6. Open a browser.
7. View a Web page with the Sitemap server control in it.
8. The sitemap now appears on your site.



Dreamweaver Extension Support

Ektron CMS400.NET has a custom Macromedia® Extension Package, which allows you to not only create your templates in Dreamweaver® or UltraDev™, but also to insert the custom functions through Dreamweaver®.

To do this, all you need is Ektron CMS400.NET, Dreamweaver®/UltraDev™, and to keep reading.

Ektron supports Dreamweaver extensions for the following scripting languages:

- ASP.NET - for .NET pages
- ASP - for Active Server Pages
- CFM - for ColdFusion pages
- JSP - Java Server Pages
- PHP - Hypertext Preprocessor

The following sections are contained in this chapter:

- ["Requirements" on page 238](#)
- ["Preparing Dreamweaver for Ektron CMS400.NET Functions" on page 238](#)
- ["Installing the Extensions" on page 239](#)
- ["Setup" on page 243](#)
- ["Editing and Saving Ektron CMS400.NET Content in Dreamweaver" on page 247](#)
- ["Using the Ektron CMS400.NET Extensions" on page 255](#)
- ["Inserting Custom Functions" on page 266](#)

Requirements

To use a Dreamweaver extension supplied by Ektron, you need to have Dreamweaver 7.0.1 or higher installed, as well as Ektron CMS400.NET.

Preparing Dreamweaver for Ektron CMS400.NET Functions

To have **Ektron CMS400.NET** function symbols appear in Dreamweaver, while editing a Web page, you must edit two `PreviewModeCustomTags.edml` Dreamweaver files. The steps below explain how to edit the files.

1. In Windows Explorer navigate to the following location

`C:\Program Files\Macromedia\Dreamweaver X\Configuration\Translators\ASP.NET_VB\`

NOTE In the path above, X represents the version of Dreamweaver you are using. For example, Dreamweaver 8 or Dreamweaver MX.

2. Open the `PreviewModeCustomTags.edml` file using Notepad.

3. Search for the following line of code in the file.

```
<searchPattern requiredLocation="openTag"><![CDATA[/runat\s*=\s*"?server"?/i]]></searchPattern>
```

4. Add the following line after it.

```
<searchPattern requiredLocation="tagName"><![CDATA[/^(?!cms:)/i]]></searchPattern>
```

WARNING! This line must be inserted after the line above. Order is important.

5. Save and close the file.

6. Navigate to the following location.

C:\Program Files\Macromedia\Dreamweaver X\Configuration\Translators\ASP.NET_Csharp\

NOTE In the path above, X represents the version of Dreamweaver you are using. For example, Dreamweaver 8 or Dreamweaver MX.

7. Open the `PreviewModeCustomTags.edml` file using Notepad.

8. Search for the following line of code in the file.

```
<searchPattern requiredLocation="openTag"><![CDATA[/runat\s*=\s*"?server"?/i]]></searchPattern>
```

9. Add the following line after it.

```
<searchPattern requiredLocation="tagName"><![CDATA[/^(?!cms:)/i]]></searchPattern>
```

WARNING! This line must be inserted after the line above. Order is important.

10. Save the file.

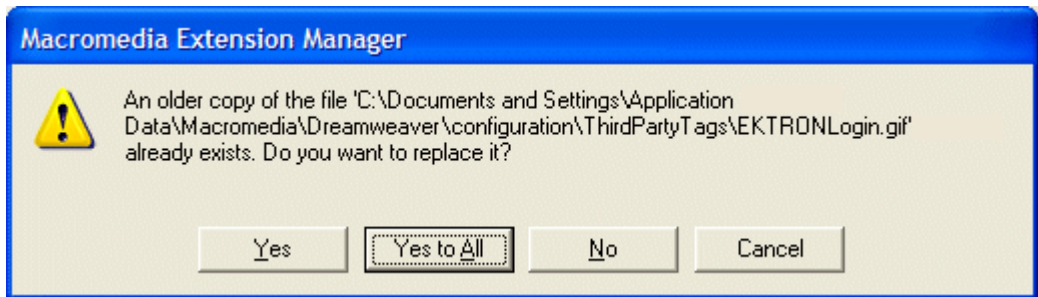
You are now ready to Install the **Ektron CMS400.NET** extension. See Also: ["Installing the Extensions" on page 239](#)

Installing the Extensions

There are two extensions that can be installed on your server:

- Ektroncms400.mxp - Adds ASP.NET extensions
- C400_ASP_PHP_CF_JSP.mxp - adds ASP, PHP, CF and JSP extensions

You can have both extensions installed at the same time. The extensions do not affect each other. When you install the second extension, you might receive the following message:



If you see this message or one that say the files are newer, click **Yes to All** to continue. Overwriting the files will not affect the extensions.

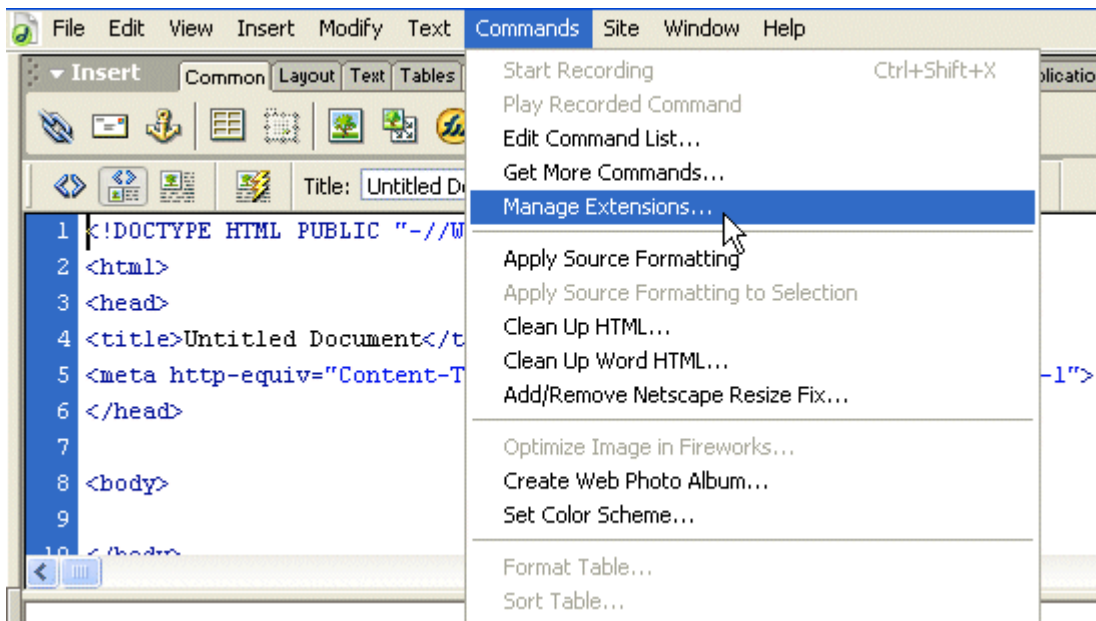
Before you can insert the Ektron CMS400.NET custom functions through Dreamweaver®, you must install at least one of the extensions. Listed below are the steps to install an extension.

NOTE For additional information about installing an extension, refer to your Dreamweaver® Manual.

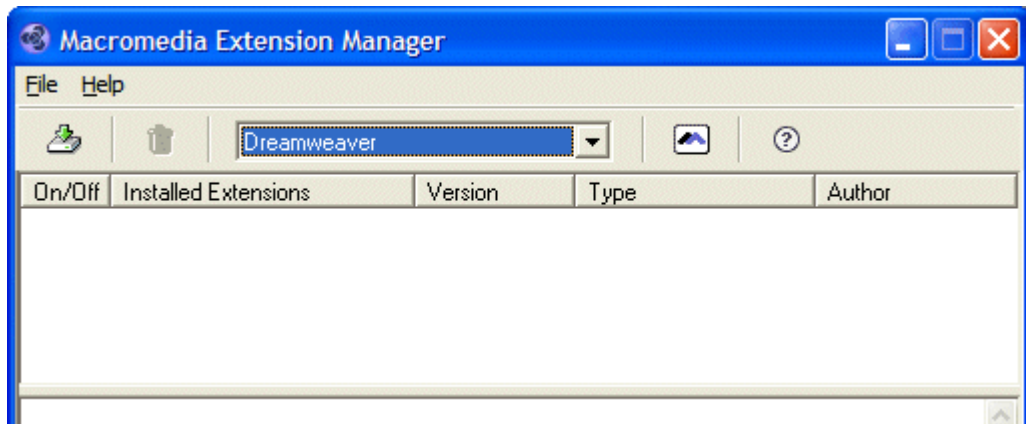
To install an extension, follow these steps.

1. Begin by following the path in Dreamweaver:

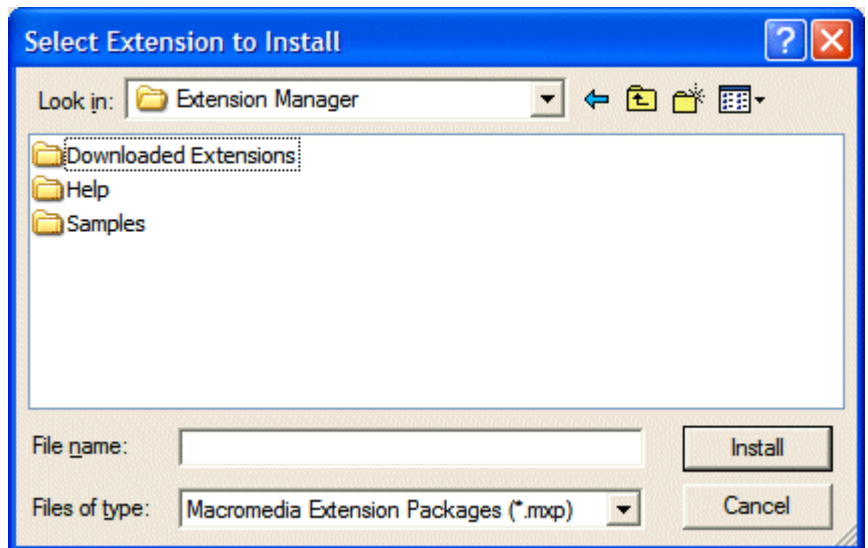
Commands > Manage Extensions...



2. The Macromedia Extension Manager is displayed.

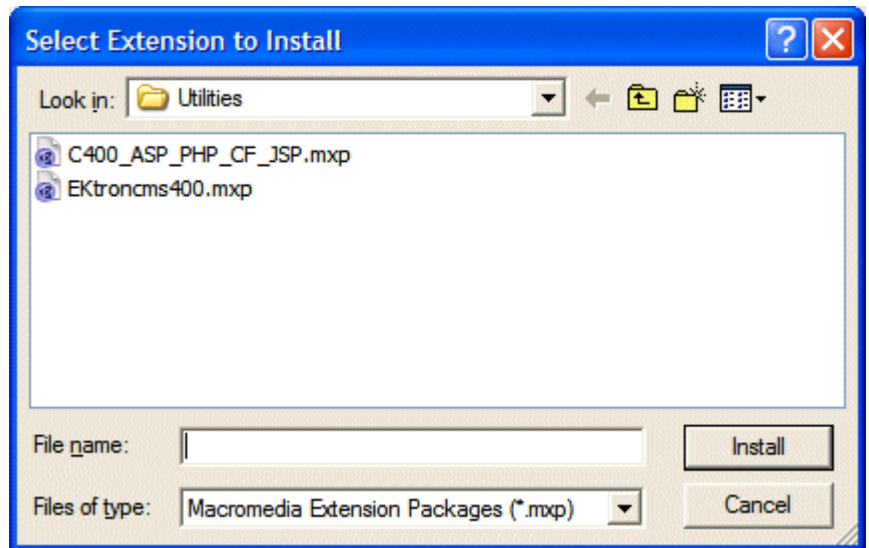


3. From this dialog box, follow the path:
File > Install Extension...
4. The Select Extension to Install window opens.



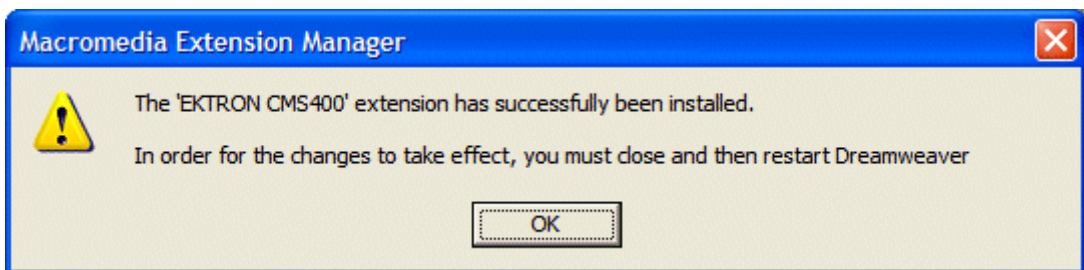
5. Locate the Macromedia® Extension Package (.mxp) file that corresponds to Ektron CMS400.NET. By default, the file gets installed to the following directory:

C:\Program Files\Ektron\CMS400vXX\Utilities



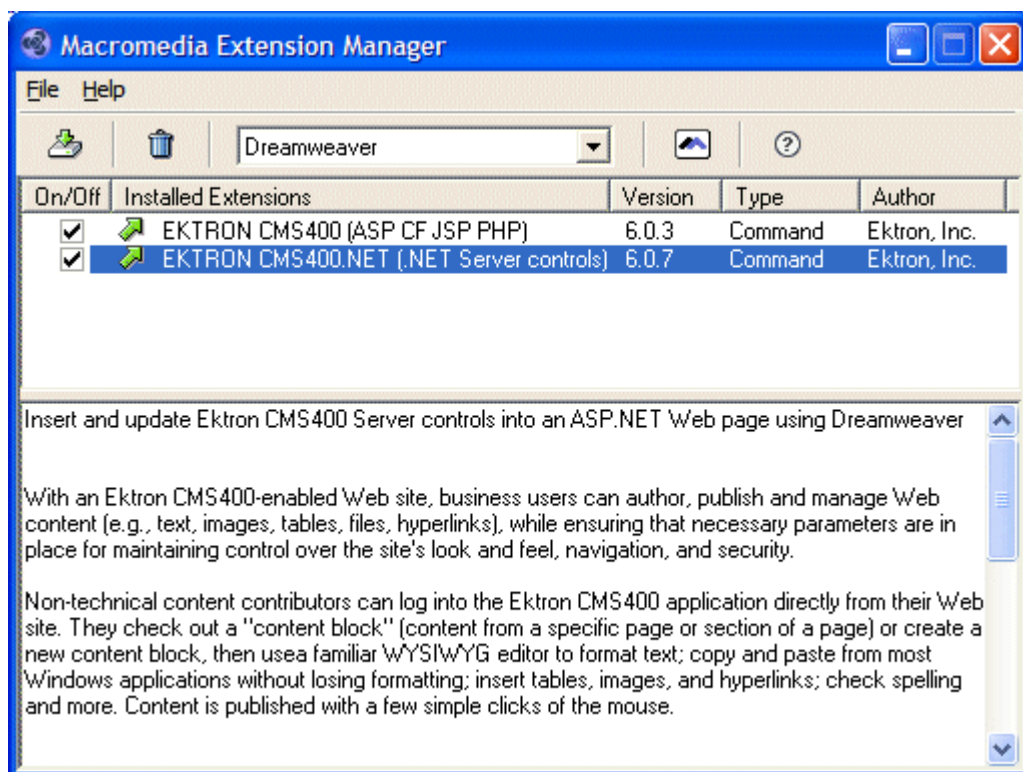
NOTE In the folder path, vXX represents the version of **Ektron CMS400.NET** you have installed.

6. After you select the file, click the **Install** button.
7. The installation begins with the Macromedia® Extensions Disclaimer. Click **Accept**.
8. The extension begins to install. Once installed, you see a confirmation box.



9. Click **OK** to continue.

10. The Macromedia Extension Manager is again displayed, but with the added extension.



11. If needed, exit Dreamweaver®, then start it back up.

Now you can use the Ektron CMS400.NET Extension Pack.

Refer to ["Using the Ektron CMS400.NET Extensions" on page 255](#) for more information on using Dreamweaver® to create and edit templates for Ektron CMS400.NET.

Setup

Before you can use Dreamweaver®/UltraDev™ to create and edit Ektron CMS400.NET templates, you should configure the Ektron CMS400.NET Extension. Listed below are the configurative options for the extension in Dreamweaver:

- Username
- Password
- Domain
- Server File
- Server

- Include
- Check for Include

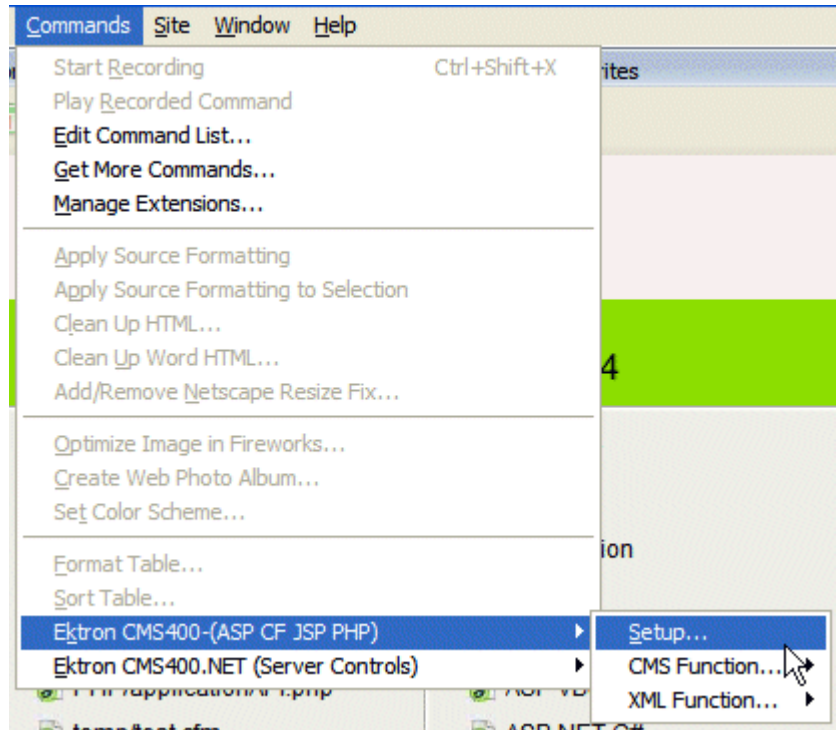
To change these options, follow these steps.

1. If you use ASP, ColdFusion, JSP or PHP, follow this path:

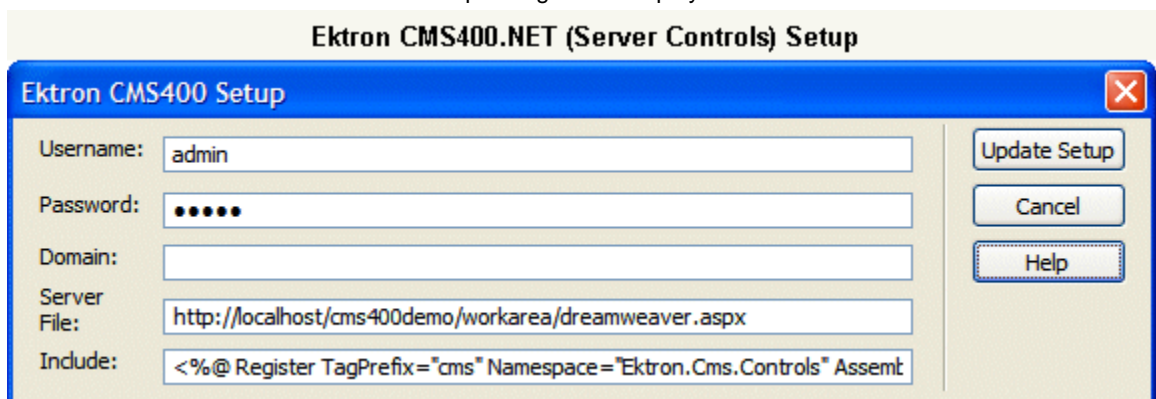
Commands > Ektron CMS400 - (ASP CF JSP PHP) > Setup...

If you use ASP.NET, follow this path:

Commands > Ektron CMS400.NET (Server Controls) > Setup...



2. The Setup dialog box is displayed on the screen.



Ektron CMS400 (ASP CF JSP PHP) Setup

Ektron CMS400-Platform Setup ✖

Username:

Password:

Domain:

Server File:

Server: ☒ ASP ☐ JSP ☐ CF ☐ PHP

Include:

☒ Check for include (check is default)

Update Setup

Cancel

Help

- Use the following table to help you update your setup information. This table refers to Ektron CMS400.NET (Server Controls) setup. For Ektron CMS400 - (ASP CF JSP PHP), see "[Ektron CMS400 - \(ASP CF JSP PHP\) Dreamweaver Setup Table.](#)" on page 246

Ektron CMS400.NET(Server Control) Dreamweaver Setup Table.

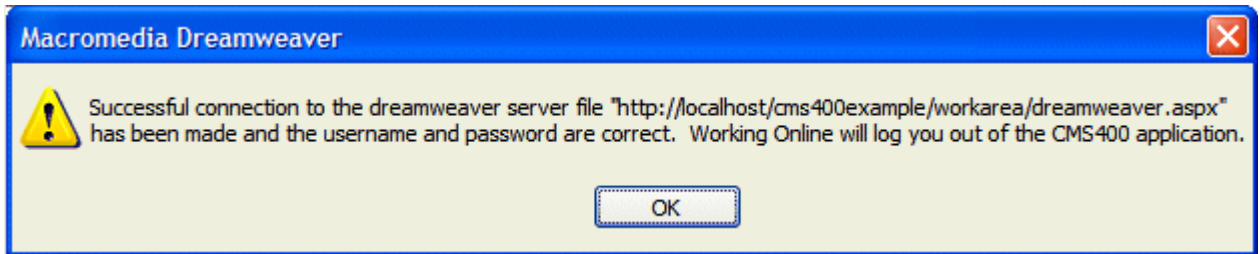
Field	Description
Username	Valid username of a user of your Ektron CMS400.NET Web site. When using the extension, the username is used to log in to the database to retrieve information about content blocks and other Web site information
Password	Enter the password for the user.
Domain	Only required if Active Directory has been enabled for your Web site. Enter the domain corresponding to the username entered.
Server File	Specify the location and file name for the dreamweaver.aspx file needed for the extension to properly work and make connection with your Ektron CMS400.NET database.
Include	Each Ektron CMS400.NET template requires an include file to point to the API with the definitions for the custom functions. Here, specify the include file.

Ektron CMS400 - (ASP CF JSP PHP) Dreamweaver Setup Table.

Field	Description
Username	Valid username of a user of your Ektron CMS400.NET Web site. When using the extension, the username is used to log in to the database to retrieve information about content blocks and other Web site information
Password	Enter the password for the user.
Domain	Only required if Active Directory has been enabled for your Web site. Enter the domain corresponding to the username entered.
Server File	Specify the location and file name for the dreamweaver.aspx file needed for the extension to properly work and make connection with your Ektron CMS400.NET database.
Server	<p>Choose the language of the server.</p> <ul style="list-style-type: none"> • ASP - for Active Server Pages • CFM - for ColdFusion pages • JSP - Java Server Pages • PHP - Hypertext Preprocessor <hr/> <p>Note: If you want to use a different language, you need to run this setup again and choose a different language.</p> <hr/>
Include	<p>Each template requires an include file to point to the API with the definitions for the custom functions. Here, specify the include file.</p> <ul style="list-style-type: none"> • For ASP use: <code><!-- #include file="applicationAPI.asp" --></code> • For CFM use: Leave blank • For JSP use: <code><%@ include file="applicationAPI.jsp" %></code> • For PHP use: <code><?php include "applicationAPI.php"; ?></code>
Check for Include	<p>If checked the Include statement is added to your page. Checked is the default.</p> <p><input checked="" type="checkbox"/> Checked - include statement added to your page</p> <p><input type="checkbox"/> Unchecked - include statement is not added to your page</p>

4. Click Update Setup.

-
5. A message appears.



NOTE A message other than this is probably due to an incorrect server file, username, or password. Recheck your settings and try again.

6. Click **OK** to continue.
7. The setup dialog box closes. You are ready to create templates using Macromedia Dreamweaver.

Editing and Saving Ektron CMS400.NET Content in Dreamweaver

Making the process of setting up and managing your Web site easier, the Ektron CMS400.NET Dreamweaver extensions allow you to perform content related tasks directly from Macromedia Dreamweaver saving you time and effort.

Through Dreamweaver, you can perform the following tasks:

- Add new content blocks to Ektron CMS400.NET
- Edit existing Ektron CMS400.NET content block

The following sections explain how to perform these tasks.

NOTE The user specified in the setup information for the extension will only be able to perform tasks they have permissions to in Ektron CMS400.NET.

Accessing Manage Content Commands...

To access the CMS Manage commands menu, follow this path:

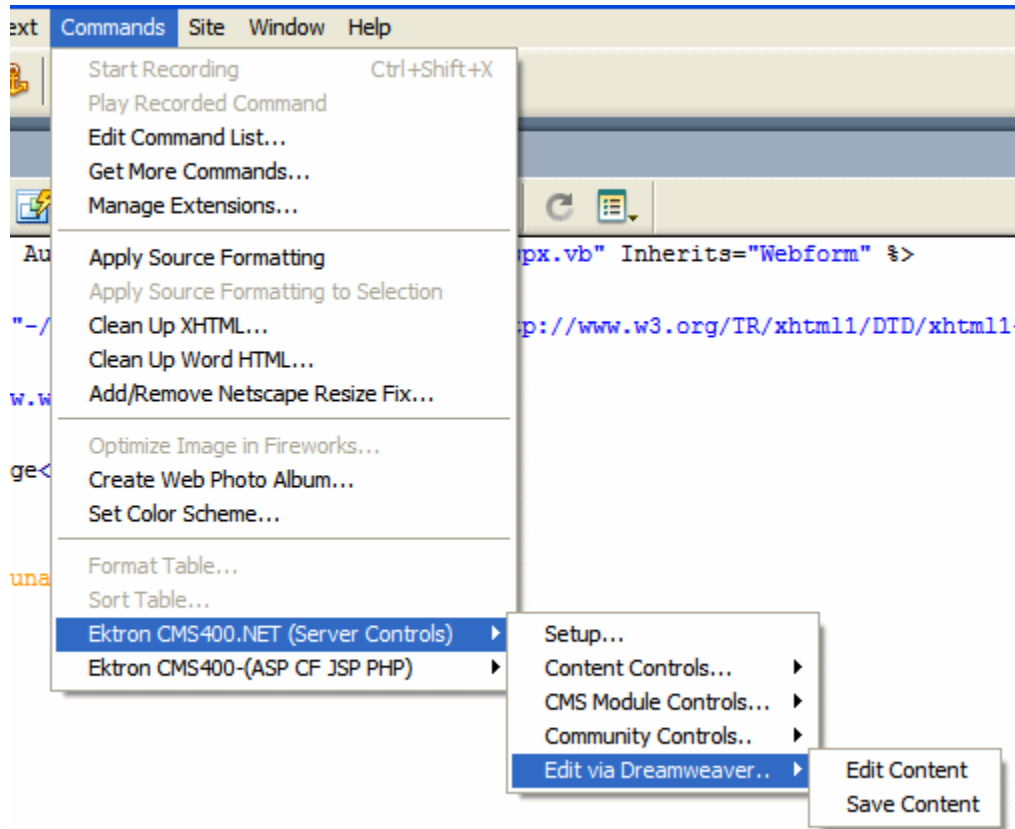
Commands > Ektron CMS400.NET (Server Controls) > Edit via Dreamweaver...

Or

Commands > Ektron CMS400 - (ASP CF JSP PHP) > Edit via Dreamweaver...

The path you follow depends on which Dreamweaver Extension you installed. If you installed both extensions, you can use either one.

A list of all CMS Manage Content commands that can be performed is displayed.



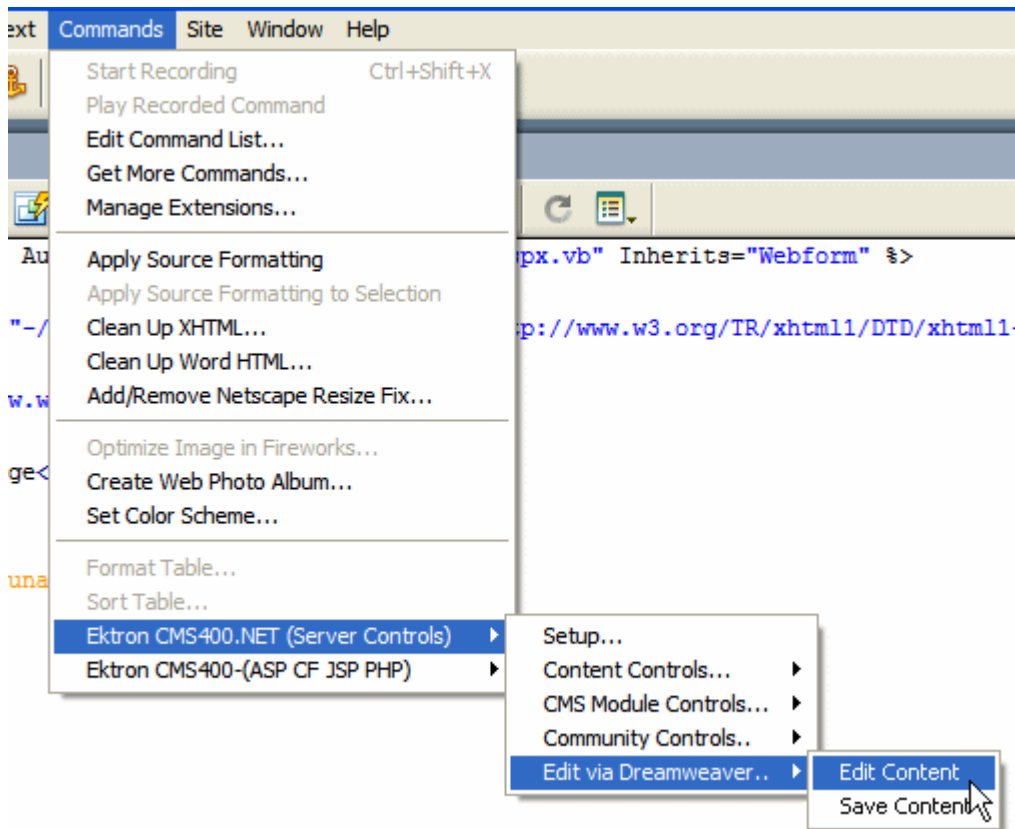
The following sections explain how to use the manage content commands.

Edit Content

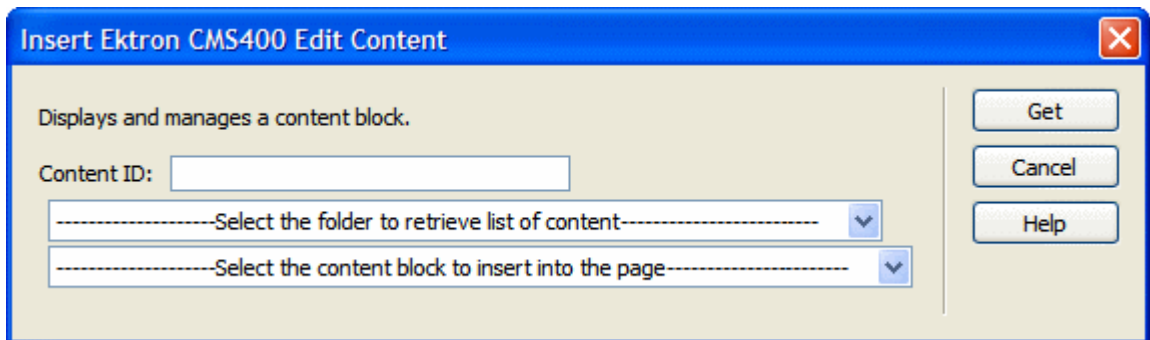
The edit content command allows you to get a content block from the Ektron CMS400.NET Web site, and edit it within Dreamweaver.

To edit a content block within Dreamweaver, perform the following steps:

1. Create a new HTML document in Dreamweaver.
2. In the Ektron CMS400.NET Manage Content menu, click on Edit Content.

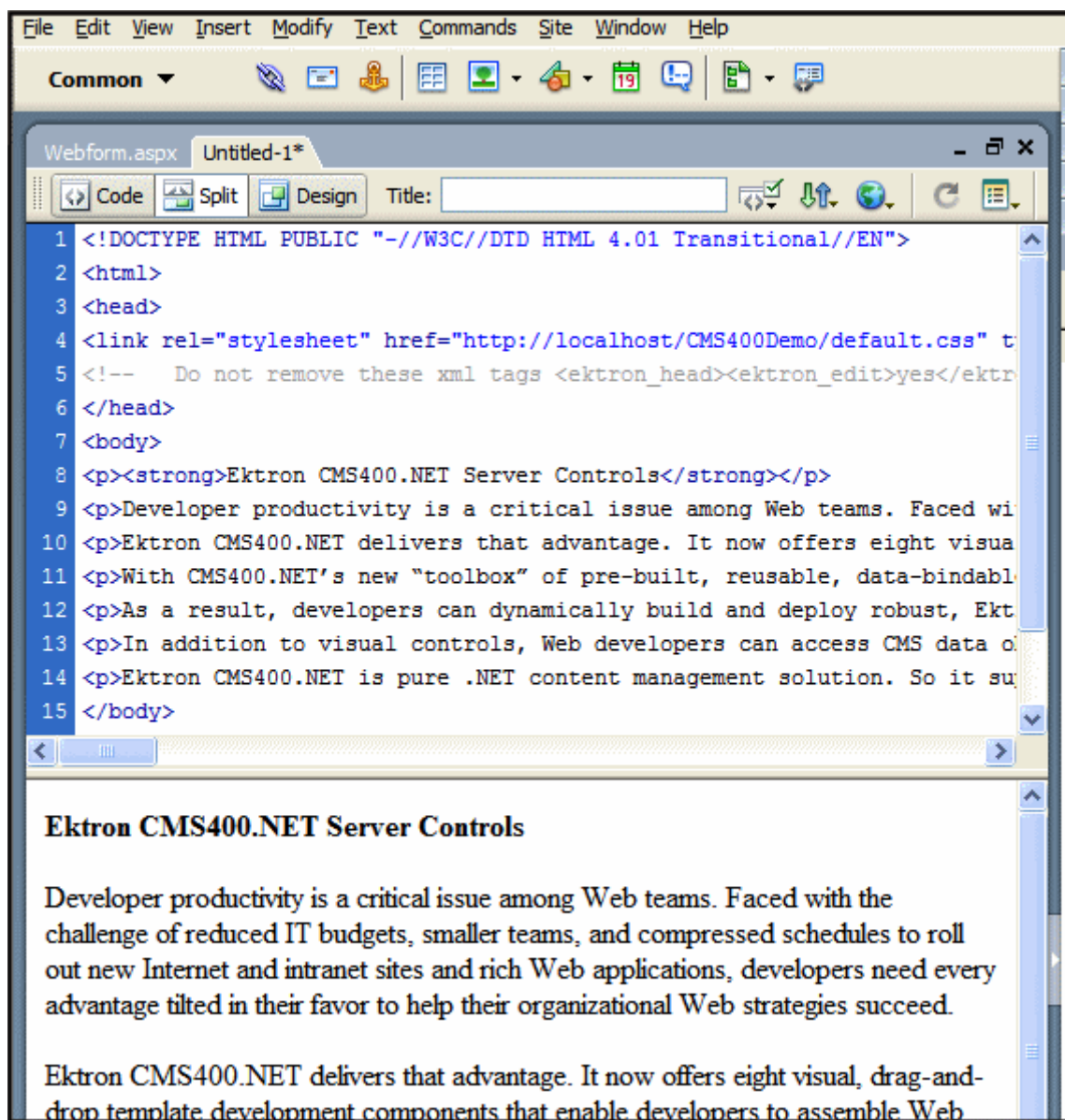


The Insert Content tag dialog box is displayed.



3. Choose the Ektron CMS400.NET content block you would like to edit by:
 - Typing the ID number of the content
 - or
 - Using the drop down lists to locate the content folder and content block

4. Click **Get** when you have chosen the content block.
The content block is inserted into Dreamweaver.



5. Make the necessary changes to the content block using Dreamweaver's editing capabilities.
See ["Saving Existing Ektron CMS400.NET Content" on page 254](#) for information about saving the content back to Ektron CMS400.NET.

Save Content

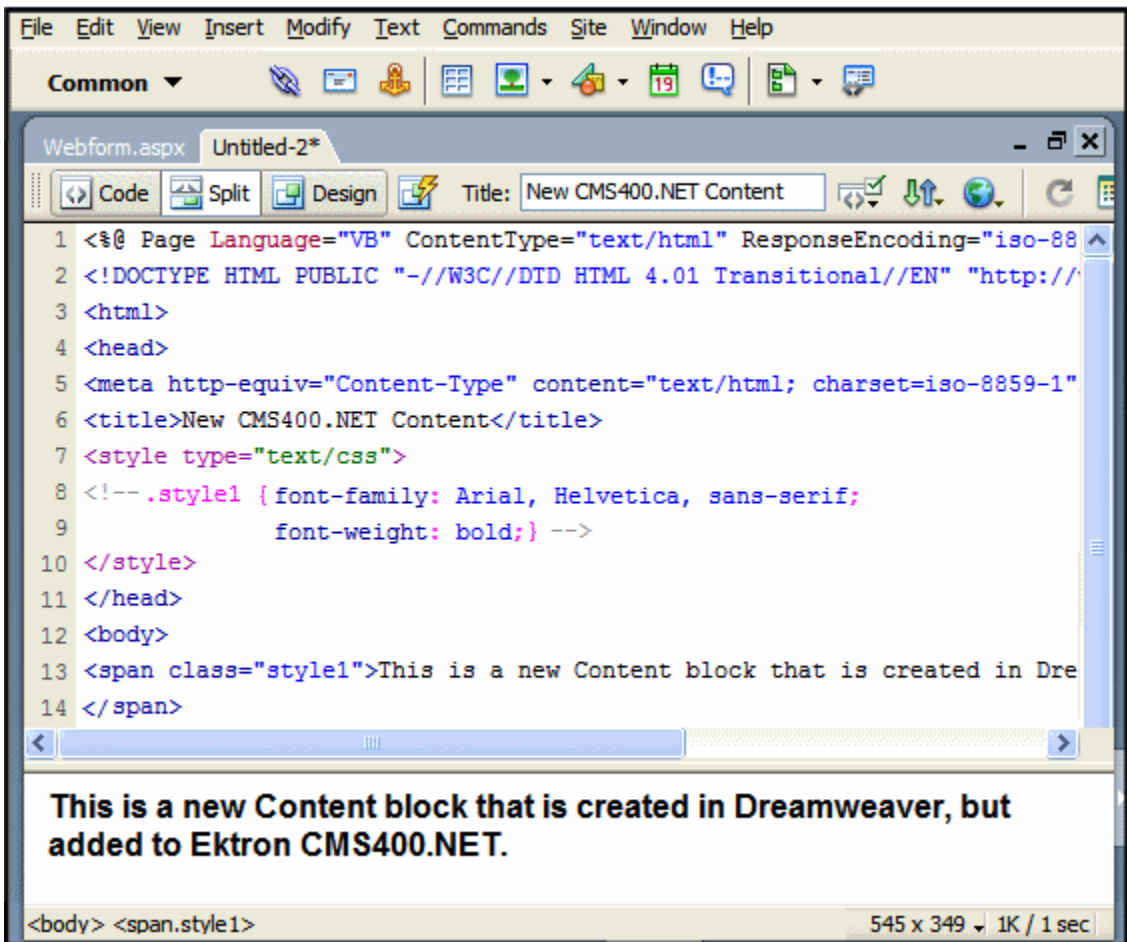
Content that has been created, or edited, in Dreamweaver can be saved to your Ektron CMS400.NET Web site by using the Save Content command.

The following section explains how to save content created, and edited, to Ektron CMS400.NET.

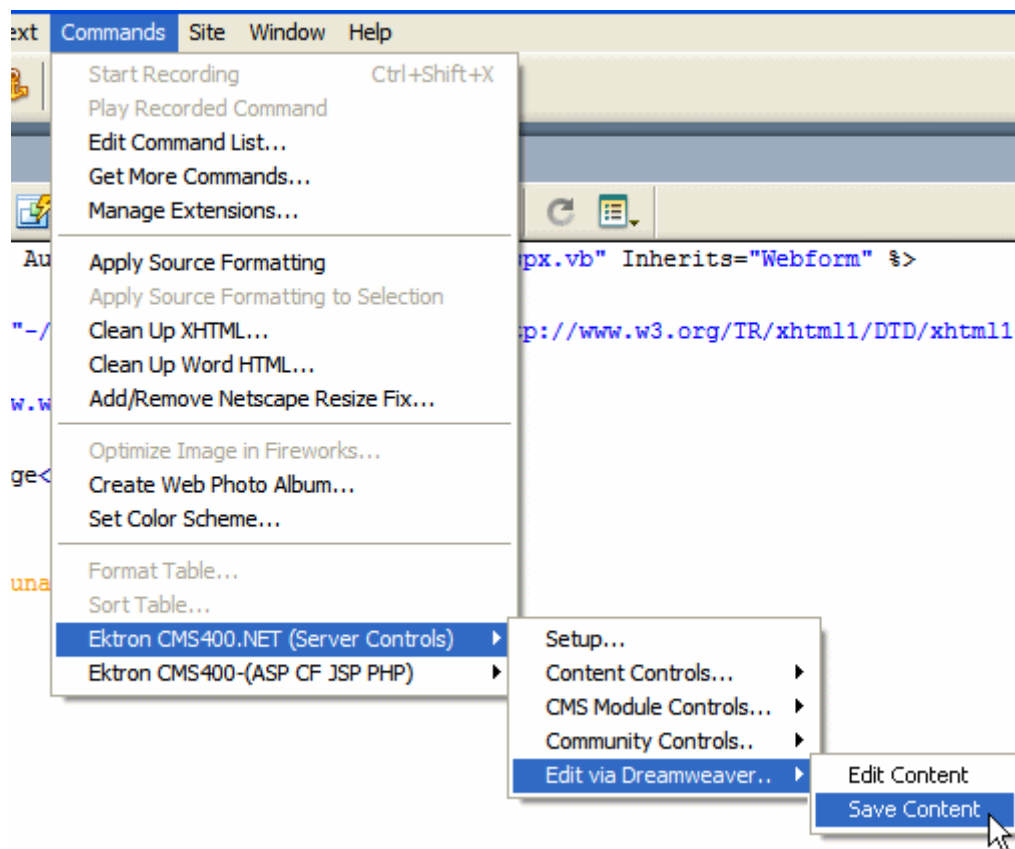
Saving New Content

To save a content block that was created in Dreamweaver, perform the following steps:

1. If you haven't already, create the new content in Dreamweaver.



2. In the Ektron CMS400.NET Manage Content menu, click on **Save Content**.



The Save Content dialog box is displayed.

A screenshot of the 'Insert Ektron CMS400 Save Content' dialog box. The dialog has a title bar with a close button. It contains four text input fields: 'Title:', 'Comment:', 'Start Date:', and 'End Date:'. Below these is a dropdown menu with the text '-----Select the folder to write content-----'. On the right side of the dialog, there are three buttons: 'Save', 'Cancel', and 'Help'.

3. Using the following table, enter the necessary information to complete the form.

Field	Description
Title	Type a unique title for the content block.
Comment	Enter a comment for the content block.
Start Date	If desired, type in a start date for the content block. NOTE: Date format MUST match what is set for the ek_dateFormat element, in the Web.config file, located in the site's webroot. By default, this is dd-mmm-yyyy hh:mm:ss tt (for example, 24-Dec-2005 10:15:00 AM)
End Date	If desired, type in an end date for the content block. NOTE: Date format MUST match what is set for the ek_dateFormat element, in the Web.config file, located in the site's webroot. By default, this is dd-mmm-yyyy hh:mm:ss tt (for example, 24-Dec-2005 10:15:00 AM)
Content Folder	Select the content folder you would like to save the new content block to.

Here is an example of this form filled out.

Insert Ektron CMS400 Save Content

Title:

Comment:

Start Date:

End Date:

▼

Save
Cancel
Help

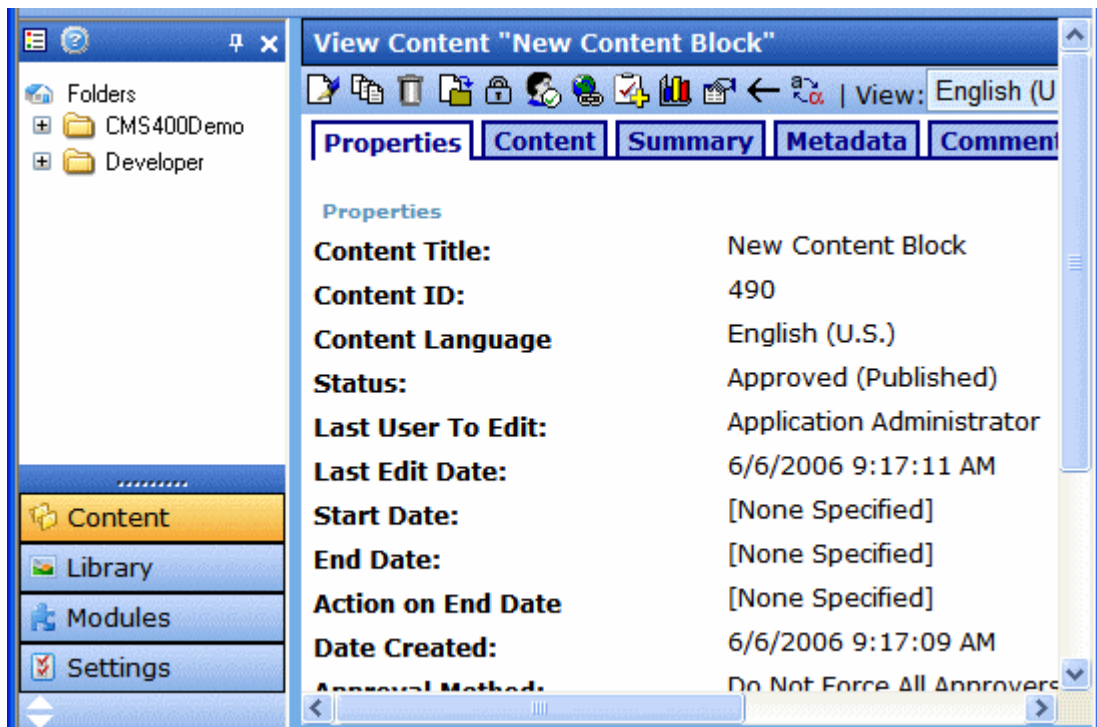
- Click **Save** to save and add the new content block the selected content folder in Ektron CMS400.NET.

The content block is added, and the following confirmation message is displayed.



NOTE If you receive a message other than the one displayed above, check your settings and try again.

Here is the new content block in the Ektron CMS400.NET Workarea.



- Once the content has been saved to Ektron CMS400.NET, you can close the page in Dreamweaver.

Saving Existing Ektron CMS400.NET Content

When Ektron CMS400.NET content has been edited in Dreamweaver, you can use the save content command to save your changes back to Ektron CMS400.NET.

To save existing Ektron CMS400.NET content, perform the following steps.

1. Make changes to an existing Ektron CMS400.NET content block as described in ["Edit Content" on page 248](#).
2. In the Ektron CMS400.NET Manage Content Menu, click on **Save Content**.
3. The Save Ektron Content dialog box is displayed.
4. Use the following table to complete the form.

Field	Description
Title	Editing this field will rename the content block in Ektron CMS400.NET.
Comment	Enter a comment for the content block.
Start Date	<p>If desired, type in a start date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set for the <code>ek_dateFormat</code> element, in the <code>Web.config</code> file, located in the site's webroot. By default, this is dd-mmm-yyyy hh:mm:ss tt (for example, 24-Dec-2005 10:15:00 AM)</p> <hr/>
End Date	<p>If desired, type in an end date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set for the <code>ek_dateFormat</code> element, in the <code>Web.config</code> file, located in the site's webroot. By default, this is dd-mmm-yyyy hh:mm:ss tt (for example, 24-Dec-2005 10:15:00 AM)</p> <hr/>
Content Folder	Disabled.

5. Click **Save** to save the changes made to the content.
The content is saved, and a confirmation message is displayed.

NOTE If you receive a message other than one confirming your content has been saved, check your settings and try again.

Using the Ektron CMS400.NET Extensions

Once the Macromedia® Extension Package has been installed for Dreamweaver, you can create and edit templates for Ektron CMS400.NET.

The following subsections are contained in this section:




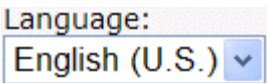

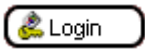
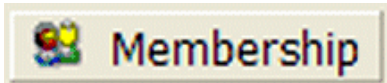

- ["Custom Function Symbols" on page 256](#)
- ["Ektron CMS400.NET Command Menus" on page 260](#)







Custom Function Symbols

During the process of inserting functions, symbols are used to represent the custom display functions. The following table explains each symbol and for which scripting languages the functions are available.

Symbol	Represents	ASP.NET	ASP - CF JSP - PHP	More Information
 CMS400 Analytic	Analytics	✓		"Analytics" on page 266
 CMS400 BLOG	Blog	✓		"Blog" on page 269
 BLOG	Blog Calendar	✓		"Blog Calendar" on page 272
 BLOG Categories	Blog Categories	✓		"Blog Categories" on page 275
 BLOG Entries	Blog Entries	✓		"Blog Entries" on page 278
 BLOG Post	Blog Posts	✓		"Blog Post" on page 281
 BLOG RecentPosts	Blog Recent Posts	✓		"Blog Recent Posts" on page 284
 BLOG Roll	Blog Roll	✓		"Blog Roll" on page 287
 BLOG RSS	Blog RSS	✓		"Blog RSS" on page 290
 Ektron > Bread > Crumb	BreadCrumb	✓		"BreadCrumb" on page 293

Symbol	Represents	ASP.NET	ASP - CF JSP - PHP	More Information
	Calendar	✓	✓	"Calendar" on page 299
	Collection	✓	✓	"Collection" on page 302
	Content Block	✓	✓	"Content Block" on page 306
	Content List	✓		"Content List" on page 310
	Content Rating	✓		"Content Rating" on page 315
	Content XSLT Tag		✓	"Content XSLT Tag..." on page 317
	DHTML Menu	✓		"DHTML Menu" on page 319
	Discussion Forum	✓		"Discussion Forum" on page 326
	Display by Meta Value	✓		"Display By Meta Value" on page 328
	Dynamic Content Block		✓	"Dynamic Content Block" on page 333

Symbol	Represents	ASP.NET	ASP - CF JSP - PHP	More Information
	Folder Bread Crumb	✓		"Folder Bread Crumb" on page 336
	Form Content Block	✓	✓	"Form" on page 338
	Language API	✓		"Language API" on page 342
	Language Select	✓		"Language Select Box" on page 344
	List Summary	✓	✓	"List Summary" on page 346
	Login/Logout button	✓	✓	"Login" on page 352
	Membership	✓		"Membership Control" on page 355
No symbol appears when Metadata is inserted in a Web page.	Metadata	✓	✓	"Metadata for ASP.NET Pages" on page 359 or "Metadata Function for ASP, CF, JSP or PHP Pages" on page 362
	Multipurpose content block.		✓	"Multipurpose Content Block" on page 365

Symbol	Represents	ASP.NET	ASP - CF JSP - PHP	More Information
 CMS400 POLL	Poll	✓		"Poll" on page 367
 elektron Random Content	Random Content		✓	"Random Content" on page 370
 elektron Random Summary	Random Summary		✓	"Random Summary" on page 371
 RSS	RSS Aggregator	✓		"RSS Aggregator" on page 373
 elektron Search Tag	Search	✓	✓	"Search" on page 375
 elektron Search Display	Search display function		✓	"Search Display" on page 387
 elektron Single Summary	Single Summary		✓	"Single Summary" on page 390
 CMS400 Site Map	Site Map	✓		"Site Map" on page 392
 elektron Menu	Standard Menu	✓		"Standard Menu" on page 394
 elektron XML Index Search Tag	XML Index Search	✓		"XML Index Search" on page 397

Ektron CMS400.NET Command Menus

The following section explains how to locate Ektron CMS400.NET commands in Dreamweaver.

There are two Ektron command menus available. One menu is for ASP.NET pages and the other is used for ASP, ColdFusion, JSP and PHP pages:

- **Commands > Ektron CMS400.NET (Server Controls)**
- **Commands > Ektron CMS400 - (ASP CF JSP PHP)**

To locate a command menu in Dreamweaver, perform the following steps:

1. Open Dreamweaver.
2. In the top menu, follow one of these paths:

- If you are using ASP.NET pages, click

Commands > Ektron CMS400.NET (Server Controls)

- If you are using ASP, ColdFusion, JSP or PHP pages, click

Commands > Ektron CMS400 - (ASP CF JSP PHP)

For a list of commands in Ektron CMS400.NET (Server Controls), see ["Ektron CMS400.NET \(Server Controls\) Functions" on page 262.](#)

For a list of commands in Ektron CMS400 - (ASP CF JSP PHP), see ["Ektron CMS400 - \(ASP CF JSP PHP\) Functions" on page 265.](#)

3. The Ektron CMS400.NET commands menu is displayed.

In Ektron CMS400.NET (**Server Controls**) the functions are broken up into the following groups:

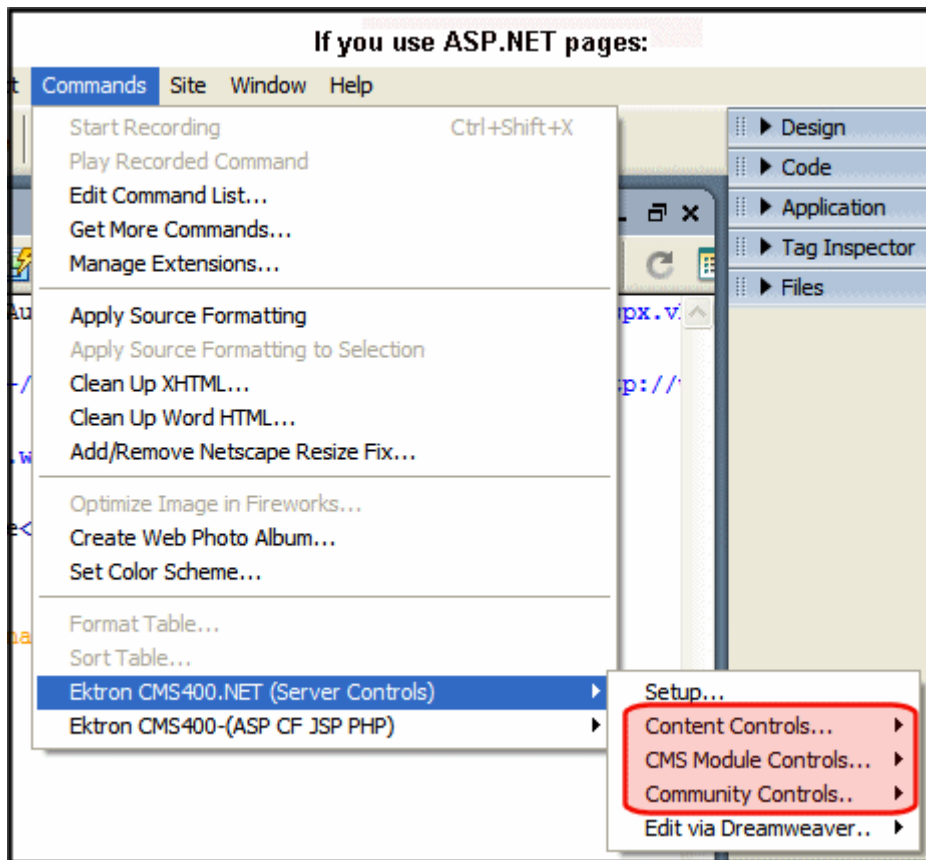
- Content Controls...
- CMS Module Controls...
- Community Controls...

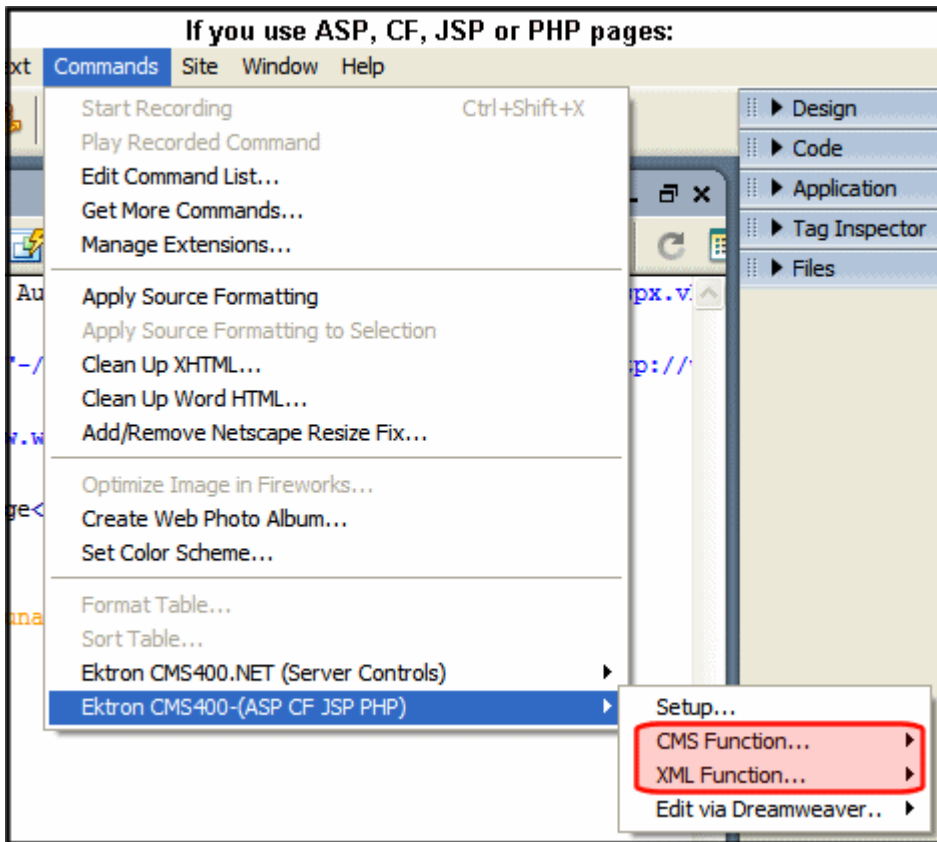
In **Ektron CMS400 - (ASP CF JSP PHP)** the functions are broken up into the following groups:

- CMS Function...
- XML Function...

NOTE For a description of the **Setup...** and **Edit via Dreamweaver...** groups, see ["Setup" on page 243](#) and ["Editing and Saving Ektron CMS400.NET Content in Dreamweaver" on page 247.](#)

Below are images showing the different views.






4. Click on the command for the task you would like to perform.
The following tables explain each command. The first table explains the "Ektron CMS400.NET (Server Controls) Functions" on page 262. The second table explains the "Ektron CMS400 - (ASP CF JSP PHP) Functions" on page 265.

Ektron CMS400.NET (Server Controls) Functions

Command	Description	More Information
Content Controls...		
Login	Inserts an Ektron CMS400.NET login function into the template.	"Login" on page 352
Content	Inserts an Ektron CMS400.NET content block function into the template.	"Content Block" on page 306
Form	Inserts an Ektron CMS400.NET form content block into the template.	"Form" on page 338

Command	Description	More Information
Poll	Displays a poll or survey created from an Ektron CMS400.NET form on a Web page.	"Poll" on page 367
Collection	Inserts an Ektron CMS400.NET collection function into the template.	"Collection" on page 302
List Summary	Inserts an Ektron CMS400.NET list summary function into the template.	"List Summary" on page 346
Metadata	Inserts an Ektron CMS400.NET metadata function into the template.	"Metadata for ASP.NET Pages" on page 359
Display By Meta Value	Inserts an Ektron CMS400.NET display by meta value block into the template.	"Display By Meta Value" on page 328
Content List	Displays a list of content blocks created from a comma delimited list.	"Content List" on page 310
Language Select	Inserts an Ektron CMS400.NET language select box into the template.	"Language Select Box" on page 344
Language API	Inserts an Ektron CMS400.NET Language API into the template.	"Language API" on page 342
ContentRating	Place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item.	"Content Rating" on page 315
Search	Inserts an Ektron CMS400.NET search function into the template.	"Search" on page 375
XML Index Search	Inserts an Ektron CMS400.NET XML Index Search into the template.	"XML Index Search" on page 397
CMS Module Controls...		
Analytics	Track statistics about visits to your Web site.	"Analytics" on page 266
Folder Bread Crumb	FolderBreadcrumbs show you the path through the sitemap to the current page.	"Folder Bread Crumb" on page 336

Command	Description	More Information
BreadCrumb	Creates a breadcrumb trail of where site visitors have been and allows them to navigate back to previous pages.	"BreadCrumb" on page 293
Calendar	Inserts an Ektron CMS400.NET event calendar function into the template.	"Calendar" on page 299
RssAggregator	Aggregates an RSS feed from an RSS Web site.	"RSS Aggregator" on page 373
Standard Menu	Inserts an Ektron CMS400.NET standard menu into the template.	"Standard Menu" on page 394
DHTML Menu	Inserts an Ektron CMS400.NET DHTML menu into the template.	"DHTML Menu" on page 319
Site Map	The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site.	"Site Map" on page 392
Community Controls...		
Membership	Adds a Membership form to your Web form. This allows site visitors to self-register themselves to your site.	"Membership Control" on page 355
Discussion Forum	Forums provide a discussion board where topics can be discussed on your Web site.	"Discussion Forum" on page 326
Blog	The Blog function allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control.	"Blog" on page 269
Blog Calendar	Display a calendar on a Web page and associate it with a blog.	"Blog Calendar" on page 272
Blog Categories	Displays the blog categories for a blog on a Web form.	"Blog Categories" on page 275
Blog Roll	Displays the blog roll for a blog on a Web form.	"Blog Roll" on page 287

Command	Description	More Information
Blog RSS	Displays the icon () for the blog's RSS feed on the Web form.	"Blog RSS" on page 290
Blog Entries	Display a list of multiple blog posts on a Web form.	"Blog Entries" on page 278
Blog Recent Posts	Displays a list of recent blog posts links on a Web form.	"Blog Recent Posts" on page 284
Blog Post	Display an individual blog post on a Web form.	"Blog Post" on page 281

Ektron CMS400 - (ASP CF JSP PHP) Functions

Command	Description	More Information
Content Controls...		
Login	Inserts an Ektron CMS400.NET login function into the template.	"Login" on page 352
Content	Inserts an Ektron CMS400.NET content block function into the template.	"Content Block" on page 306
Dynamic	Inserts an Ektron CMS400.NET dynamic content block placeholder function into the template.	"Dynamic Content Block" on page 333
Collection	Inserts an Ektron CMS400.NET collection function into the template.	"Collection" on page 302
List Summary	Inserts an Ektron CMS400.NET list summary function into the template.	"List Summary" on page 346
Metadata	Inserts an Ektron CMS400.NET metadata function into the template.	"Metadata for ASP.NET Pages" on page 359
Search	Inserts an Ektron CMS400.NET search function into the template.	"Search" on page 375
Search Display Tag	Inserts an Ektron CMS400.NET search display function into the template.	"Search Display" on page 387

Command	Description	More Information
Single Summary Tag	Inserts an Ektron CMS400.NET single summary function into the template.	"Single Summary" on page 390
Random Content Tag	Inserts an Ektron CMS400.NET random content function into the template.	"Random Content" on page 370
Random Summary Tag	Inserts an Ektron CMS400.NET random summary function into the template.	"Random Summary" on page 371
Calendar	Inserts an Ektron CMS400.NET event calendar function into the template.	"Calendar" on page 299
Form	Inserts an Ektron CMS400.NET form content block into the template.	"Form" on page 338
Multipurpose Tag	Insert a content block or form content block into the template.	"Multipurpose Content Block" on page 365
XML Function...		
Content XSLT Tag	Inserts an XML content block with your choice of XSLT in the template. <hr/> NOTE: Used to insert an XML content block with an XSLT other than the default applied to it. <hr/>	"Content XSLT Tag..." on page 317

Inserting Custom Functions

CMS Function

The CMS Function commands let you insert standard Ektron CMS400.NET display functions into your template.

The following sections explain how to insert each function.

Analytics

For a description of the Analytics function, see ["Content Analytics Server Control" on page 89](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Controls...**

To insert the Analytics function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Analytics**.
3. The Insert Ektron CMS400.NET Analytics box appears.

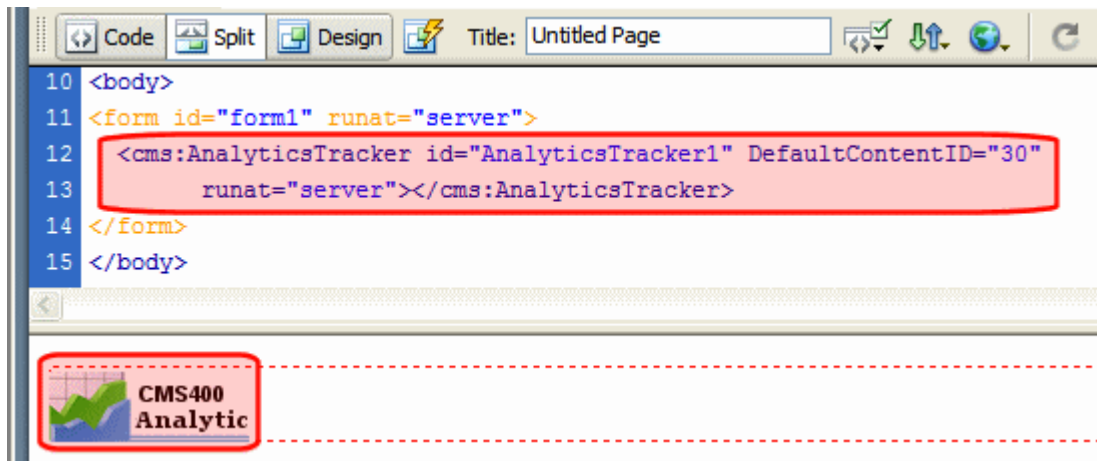
4. Use the following table to assist you with inserting the Analytics function.

Field	Description
Control ID	A unique name for the function you are inserting.
Content ID	Specify the ID number of a content item being tracked by this function. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content drop down boxes.
Dynamic Content Parameter	To make this content analytic control dynamic, select id . When you do, this server control is attached to the content block passed as a URL parameter.
Hide	<p>Used to hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

An Analytic icon appears to indicate that content is being tracked. If you are logged in as a CMS user and visit the web page, you see the Analytic information. If you are not logged in, you see nothing.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:AnalyticsTracker id="AnalyticsTracker1" DefaultContentID="30" runat="server"></cms:AnalyticsTracker></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog

For a description of the Blog function, see ["Blog Server Control" on page 43](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):


- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog**.

3. The Insert Ektron CMS400.NET Blog box appears.

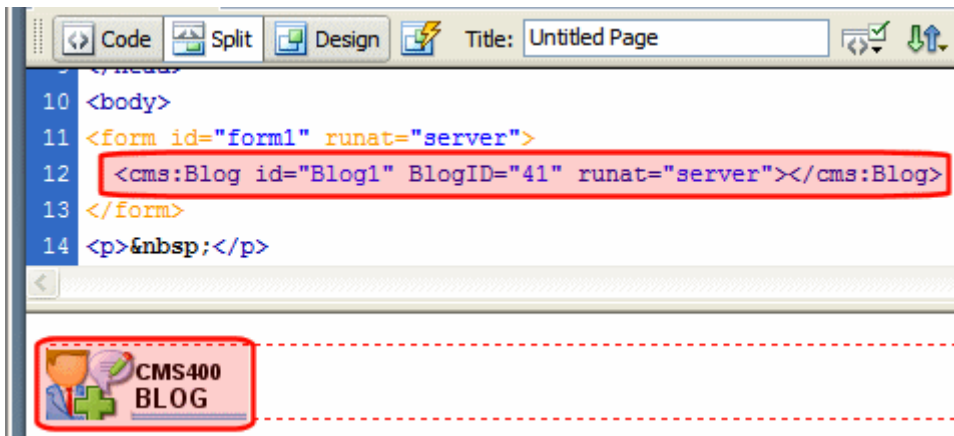
4. Use the following table to assist you with inserting the Blog function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
ShowRSS	Displays the icon for the RSS feed () when the box is checked. The default is Checked . <input checked="" type="checkbox"/> Checked = Display RSS feed <input type="checkbox"/> Unchecked = Do not display RSS feed
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). See Also: " Caching with Server Controls " on page 36.

Field	Description
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog icon appears and indicates the Blog's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:Blog id="Blog1" BlogID="41" runat="server"></cms:Blog></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Calendar

For a description of the Blog Calendar function, see ["BlogCalendar Server Control" on page 50](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Calendar function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Calendar**.

3. The Insert Ektron CMS400.NET Blog Calendar box appears.

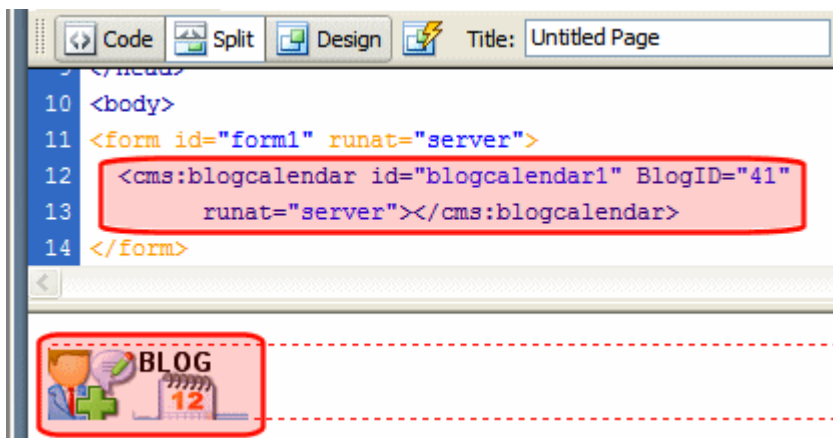
4. Use the following table to assist you with inserting the Blog Calendar function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog Calendar icon appears and indicates the Blog Calendar's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:blogcalendar id="blogcalendar1" BlogID="41" runat="server"></cms:blogcalendar></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Categories

For a description of the Blog Categories function, see "[BlogCategories Server Control](#)" on page 55 and "[Blog Server Controls](#)" on page 43.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Categories function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Categories**.

3. The Insert Ektron CMS400.NET Blog Categories box appears.

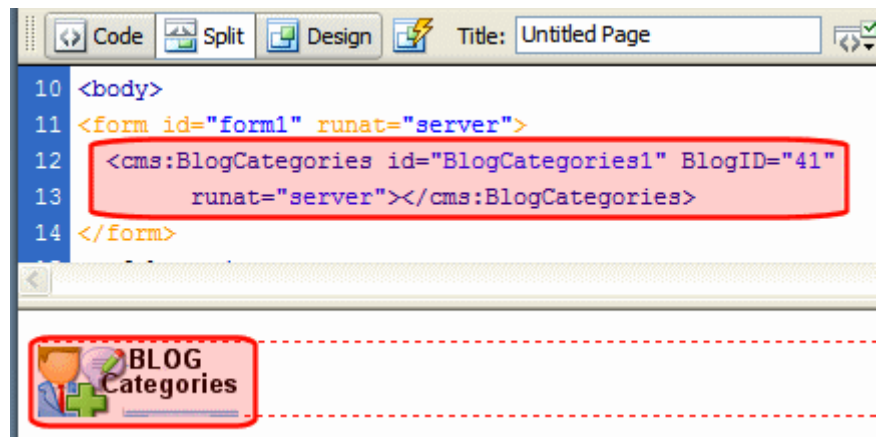
4. Use the following table to assist you with inserting the Blog Categories function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide	Used to hide a function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden. <hr/>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog Categories icon appears and indicates the Blog Categories' location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogCategories id="BlogCategories1" BlogID="41" runat="server"></cms:BlogCategories></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Entries

For a description of the Blog Entries function, see ["BlogEntries Server Control" on page 45](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Entries function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Entries**.

3. The Insert Ektron CMS400.NET Blog Entries box appears.

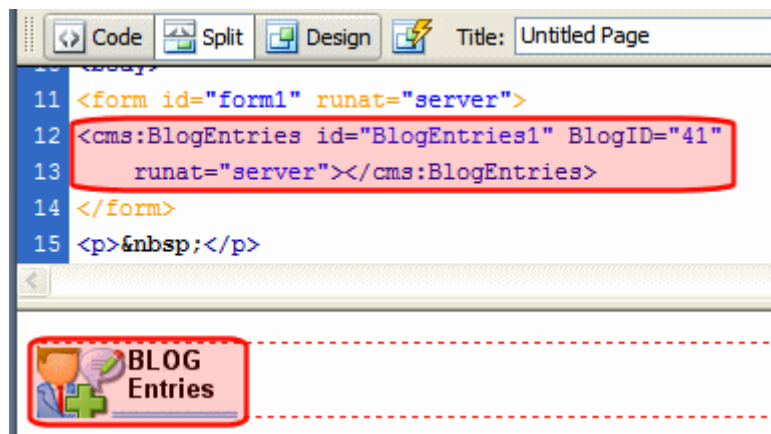
4. Use the following table to assist you with inserting the Blog Entries function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: " Caching with Server Controls " on page 36.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog Entries icon appears and indicates the Blog Entries' location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogEntries id="BlogEntries1" BlogID="41" runat="server"></cms:BlogEntries></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Post

For a description of the Blog Post function, see ["BlogPost Server Control" on page 48](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Post function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Post**.

3. The Insert Ektron CMS400.NET Blog Post box appears.

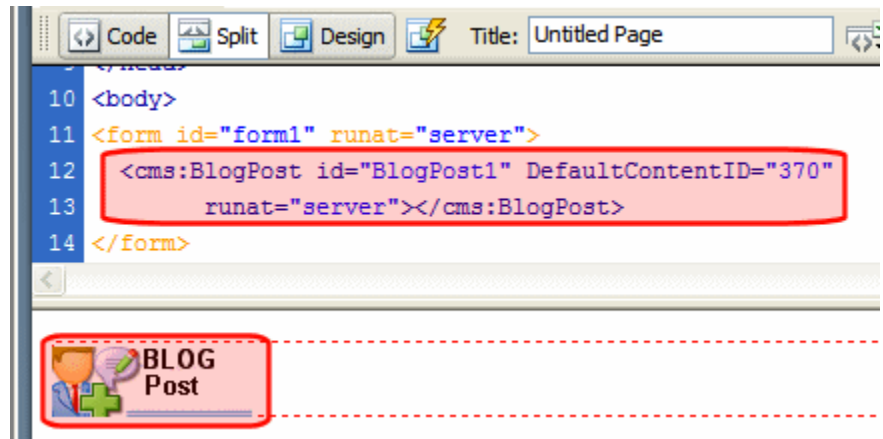
4. Use the following table to assist you with inserting the Blog Post function.

Field	Description
Control ID	A unique name for the function you are inserting.
DefaultContent ID	The ID of a blog post in CMS400.NET. If you do not know the Blog Post's ID, you can navigate to it using the Select Category and Select Content drop down boxes.
DynamicParameter	To make this blog post dynamic, select id . When you do, this server control uses the blog post passed as a URL parameter.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
# of seconds - CacheInterval	<p>Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.</p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog Post icon appears and indicates the Blog Post's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogPost id="BlogPost1" DefaultContentID="370" runat="server"></cms:BlogPost></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Recent Posts

For a description of the Blog Recent Posts function, see "[BlogRecentPosts Server Control](#)" on page 56 and "[Blog Server Controls](#)" on page 43.

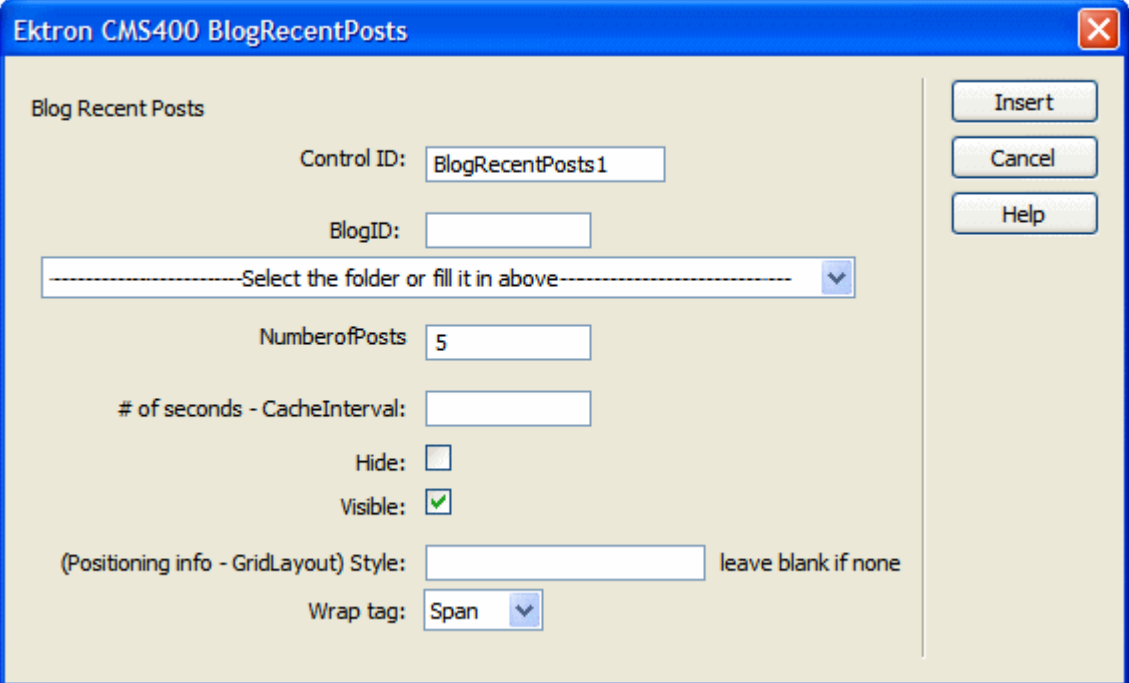
This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Recent Posts function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Recent Posts**.

3. The Insert Ektron CMS400.NET Blog Recent Posts box appears.



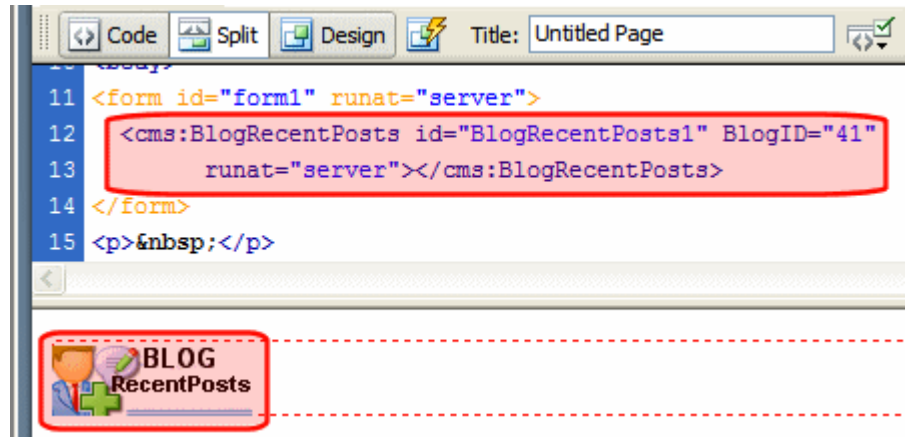
4. Use the following table to assist you with inserting the Blog Recent Posts function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
NumberOfPosts	Sets the number of post links to display.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .

Field	Description
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Blog Recent Posts icon appears and indicates the Blog Recent Posts' location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogRecentPosts id="BlogRecentPosts1" BlogID="41" runat="server"></cms:BlogRecentPosts></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog Roll

For a description of the Blog Roll function, see ["BlogRoll Server Control" on page 54](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog Roll function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog Roll**.

3. The Insert Ektron CMS400.NET Blog Roll box appears.

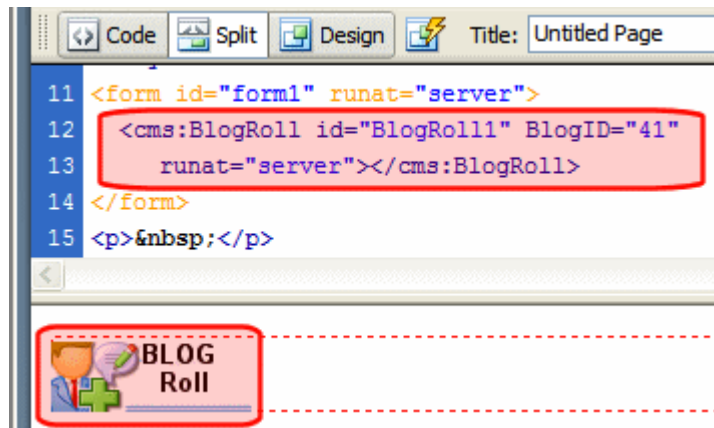
4. Use the following table to assist you with inserting the Blog Roll function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **Insert**.

A Blog Roll icon appears and indicates the Blog Roll's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogRoll id="BlogRoll1" BlogID="41" runat="server"></cms:BlogRoll></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Blog RSS

For a description of the Blog RSS function, see ["BlogRSS Server Control" on page 57](#) and ["Blog Server Controls" on page 43](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Blog RSS function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Blog RSS**.

3. The Insert Ektron CMS400.NET Blog RSS box appears.

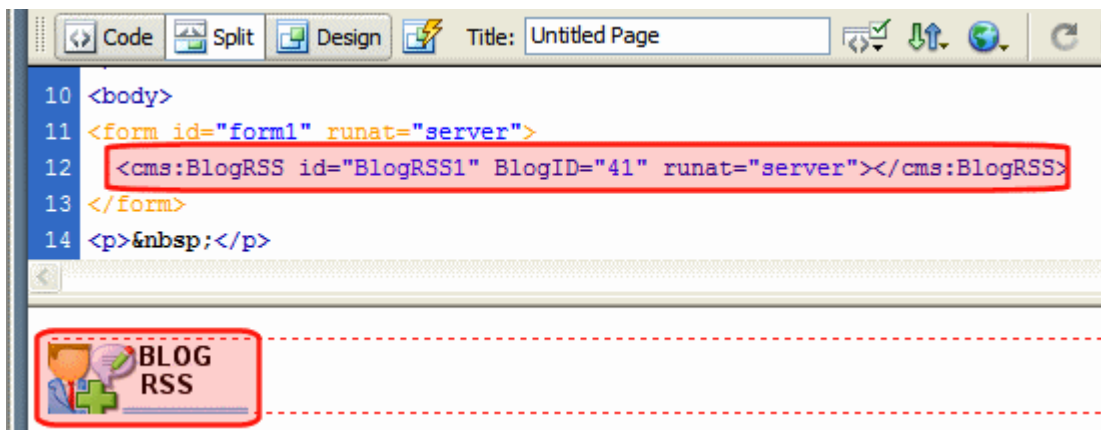
4. Use the following table to assist you with inserting the Blog RSS function.

Field	Description
Control ID	A unique name for the function you are inserting.
Blog ID	The ID of the blog in CMS400.NET. If you do not know the Blog ID, you can choose it from the Select Folder drop down box.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **Insert**.

A Blog RSS icon appears and indicates the Blog RSS's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:BlogRSS id="BlogRSS1" BlogID="41" runat="server"></cms:BlogRSS></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

BreadCrumb

For a description of the BreadCrumb function, see ["BreadCrumb Server Control" on page 59](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Controls...**

To insert a BreadCrumb Trail:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **BreadCrumb**.

3. The BreadCrumb dialog box appears.

Ektron CMS400 Bread Crumb

Used to display a Bread Crumb

Control ID: BreadCrumb1

Meta tag to use - MetadataName:

DisplayTitle:

CurrentPageIndicator:

Content ID:

-----Select the category to retrieve list of content-----

-----Select the content block to insert into the page-----

Dynamic Content Parameter: None

IconPath IconAlt:

LinkTarget _self Link Last Item:

Create Hyperlinks ☒ MaxItems: 5

Seperator > Display Style: Horizontal

Normal Mode ☒ Hide:

Visible ☒ Wrap tag: Span

(Positioning info - GridLayout) Style: leave blank if none


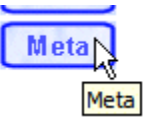
Insert

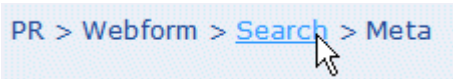
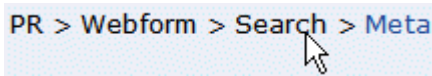

Cancel

Help

4. Use the following table to assist you with inserting a BreadCrumb Trail.

Field	Description
Control ID	A unique name for the function you are inserting.
Meta tag to use - MetadataName	Specify the name of a Metadata Type that you want to associate with the page. For more information, see "BreadCrumb Metadata Type" on page 65

Field	Description
DisplayTitle	<p>Enter text to describe this Web form when it appears in the breadcrumb trail. For example, if the Web form whose properties you are defining is used for all Human Resources pages on your site, enter Human Resources. See A/so: "BreadCrumb Properties Apply to Web Forms, Not Web Pages" on page 61</p> <p>The default is Title.</p> <p>If you define an image in the IconPath property below, the image appears in the trail, followed by this text.</p>
CurrentPageIndicator	<p>If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail. See example below.</p> <p style="text-align: center;">Dev > WebForm > pr > FAQ > Meta </p> <p>These characters appear <i>after</i> the image or title that identifies the current page in the breadcrumb trail.</p>
Content ID	Gets or Sets the ContentID for the BreadCrumb display title. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content drop down boxes.
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.
IconPath	<p>If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image.</p> <hr/> <p>Important: The image location must be relative to the Web root.</p> <hr/> <p>For example: \cms400demo\Workarea\Images\bc_meta_icon.gif</p>
IconAlt	<p>If you define an image in the IconPath property, enter any "Alt" text that should appear when a site visitor hovers the cursor over that image. Here is an example.</p> <p style="text-align: center;"></p>
LinkTarget	<p>Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.</p> <ul style="list-style-type: none"> _Self -opens in same window _Top - opens in parent window _Blank - opens in new window _Parent - opens in the parent frame

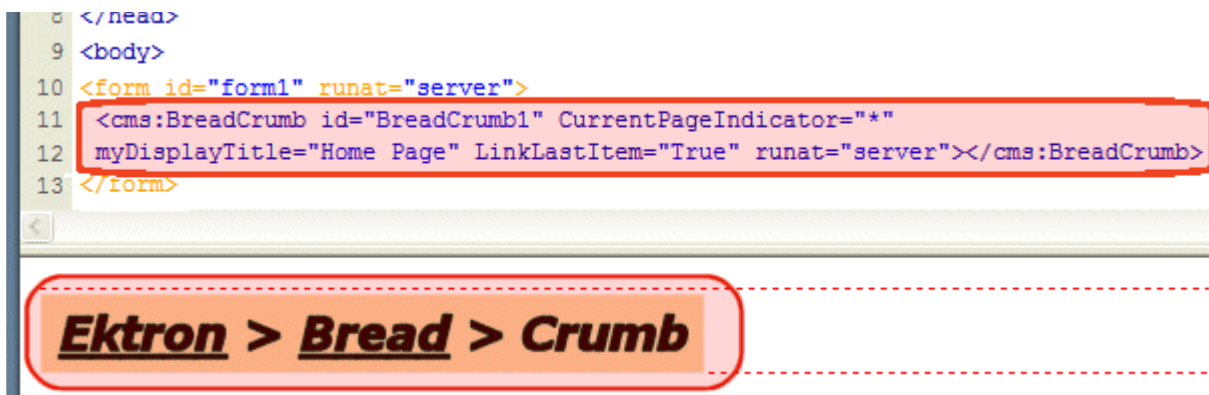
Field	Description
Link Last Item	<p>Use this property to determine whether the last breadcrumb item appears as a hyperlink on this Web form. If this property is set to true, and a user clicks the item, the current page reappears.</p> <p>True = last item is a hyperlink</p> <p>False = last item is an image and/or text only; the user cannot click on it</p>
Create Hyperlinks	<p>Unchecking this box makes the breadcrumb trail appear as non-hyperlinked plain text. The default is Checked.</p> <p><input checked="" type="checkbox"/> Checked = breadcrumb trail is hyperlinked</p> <p></p> <p><input type="checkbox"/> Unchecked = breadcrumb trail is plain text</p> <p></p>
MaxItems	<p>Enter the maximum number of items in the breadcrumb trail on this Web form. The default is 5.</p> <p>If you set a value of 1 or greater and the user navigates beyond that number of pages, only the most recent pages appear. The older pages disappear from the trail.</p> <p>To allow an unlimited number of breadcrumb trail items on this Web form, enter zero (0).</p>
Separator	<p>Enter one or more characters to separate the items in a breadcrumb trail on this Web form.</p> <p>The default character is the greater than sign (>).</p> <p></p>

Field	Description
DisplayStyle	<p>Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.</p> <p>This is an example of Horizontal.</p> <p>Dev > WebForm > pr > FAQ > Meta*</p> <p>This is an example of Vertical.</p> <p> > Dev > WebForm > pr > FAQ > Meta* </p>
Mode	<p>Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.</p> <p>Normal = breadcrumb trail is hyperlinked</p> <p>PR > Webform > Search > Meta</p> <p>DisplayOnly = breadcrumb trail is plain text</p> <p>PR > Webform > Search > Meta</p>
Hide	<p>Used to hide a Language API in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide</p> <p><input type="checkbox"/> Unchecked = Show Language API</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

- When completed, click **Insert**.

A graphic defines where the BreadCrumb Trail appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<pre><cms:BreadCrumb id="BreadCrumb1" CurrentPageIndicator="*" myDisplayTitle="Home Page" LinkLastItem="True" runat="server"></cms:BreadCrumb></pre>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Calendar

For a description of the Calendar function, see ["Calendar Server Control" on page 75](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- **Commands > Ektron CMS400.NET (Server Controls) > CMS Modules Controls...**

To insert an Calendar function:

NOTE Inserting a calendar function does not add a reference to a calendar style sheet. This step must be performed manually.

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click on **Calendar**
3. The Insert Calendar box is displayed.

Insert Ektron cms400 Calendar

Displays and manages a content block.

Control ID:

Content ID:

-----Select the calendar to insert into the page-----

Dynamic Content Parameter:

CalDisplayType:

ShowXML: ☐

Hide: ☐

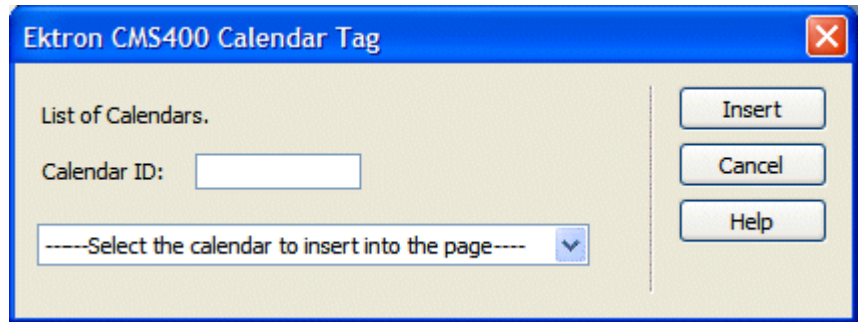
Visible: ☒

(Positioning info - GridLayout) Style: leave blank if none

Wrap tag:

IMPORTANT!

When inserting an event calendar using ASP, CF, JSP or PHP, the dialog box that appears only allows you to set the Calendar ID or choose one from the list.



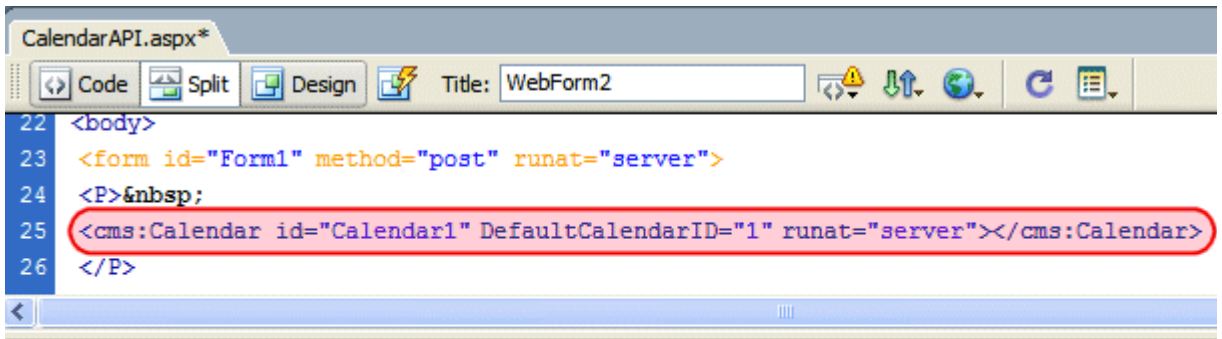
4. Use the following table to assist you with inserting a calendar.

Field	Description
Control ID <u>Used in ASP.NET only.</u>	A unique name for the function you are inserting.
Content ID	Type in the Content ID of the calendar that you want to insert. If you do not know the Content ID, you can navigate to it using the Select Calendar drop down box.
Select Calendar	The drop-down box is populated with a list of all the calendar titles. When a title is selected, the content ID textbox is filled with that content block's id number. If you insert the calendar ID, you do not need to select a calendar.
Dynamic Content Parameter <u>Used in ASP.NET only.</u>	To make this calendar dynamic, select calendar_id . When you do, this server control uses the calendar passed as a URL parameter.
CalDisplayType <u>Used in ASP.NET only.</u>	Specify the Calendar view type. The default is Month . <ul style="list-style-type: none"> • Month - A 30 day calendar month view. • Day - Single day view. • Inline - Display events that occur from the start date of the calendar through the end date. • MonthUpWithEvent - Displays a month calendar. Events appear to the right of the calendar when you click on a day.

Field	Description
ShowXML <hr/> Used in ASP.NET only. <hr/>	<p>At runtime, output the raw XML data instead of the calendar. With this XML data, you can apply your own XSL to create your own custom display formats.</p> <p>Default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Show raw XML data</p> <p><input type="checkbox"/> Unchecked = Show Calendar</p>
Hide <hr/> Used in ASP.NET only. <hr/>	<p>Used to hide a calendar in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide calendar</p> <p><input type="checkbox"/> Unchecked = Show calendar</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible <hr/> Used in ASP.NET only. <hr/>	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only. <hr/>	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag <hr/> Used in ASP.NET only. <hr/>	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **insert**.

A graphic defines where the calendar appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:Calendar id="Calendar1" DefaultCalendarID="1" runat="server"></cms:Calendar></code>
ASP	<code><% ecmEvtCalendar (1) %></code>
ColdFusion	<code><cfmodule Calendar_id="1" template="#request.ecm.AppPath#ecmEvtCalendar.cfm"></code>
JSP	<code><%= ecmEvtCalendar(1) %></code>
PHP	<code><?php ecmEvtCalendar (1); ?></code>

Collection

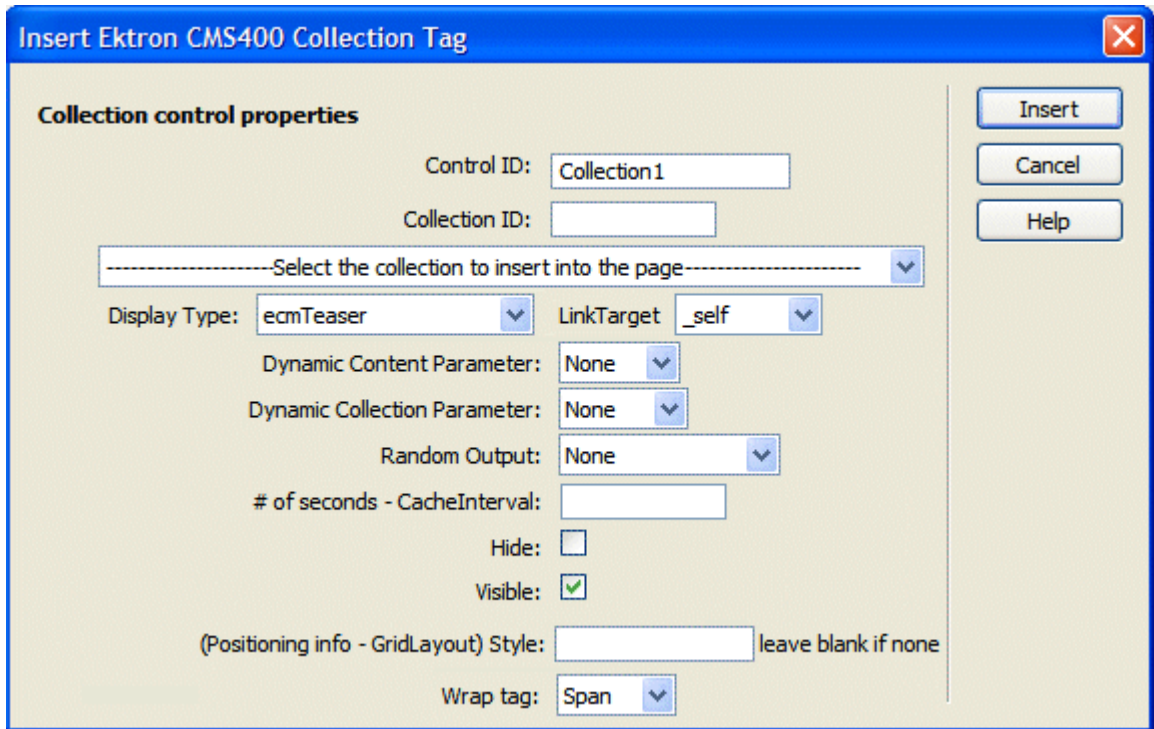
For a description of the Collection function, see ["Collection Server Control" on page 82](#)

This tag appears on the following menu(s):

- Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

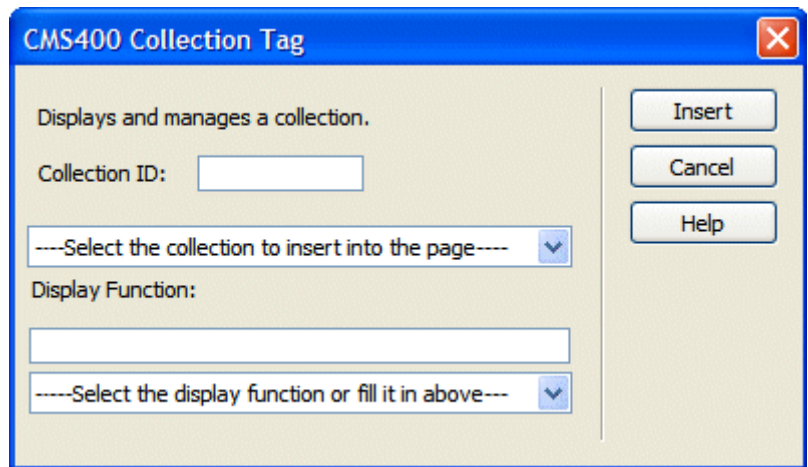
To insert a collection:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Collection**.
3. The Insert Ektron Collection box is displayed.



The dialog box is titled "Insert Ektron CMS400 Collection Tag" and features a close button (X) in the top right corner. It is divided into two main sections. The left section, titled "Collection control properties", contains several configuration options: "Control ID" with a text box containing "Collection1"; "Collection ID" with an empty text box; a dropdown menu labeled "-----Select the collection to insert into the page-----"; "Display Type" with a dropdown menu set to "ecmTeaser"; "LinkTarget" with a dropdown menu set to "_self"; "Dynamic Content Parameter" with a dropdown menu set to "None"; "Dynamic Collection Parameter" with a dropdown menu set to "None"; "Random Output" with a dropdown menu set to "None"; "# of seconds - CacheInterval" with an empty text box; "Hide" with an unchecked checkbox; "Visible" with a checked checkbox; "(Positioning info - GridLayout) Style" with an empty text box and the instruction "leave blank if none"; and "Wrap tag" with a dropdown menu set to "Span". The right section contains three buttons: "Insert", "Cancel", and "Help".

IMPORTANT! When inserting a collection using ASP, CF, JSP or PHP, a different dialog box appears.



The dialog box is titled "CMS400 Collection Tag" and features a close button (X) in the top right corner. It contains the following elements: a description "Displays and manages a collection."; "Collection ID" with an empty text box; a dropdown menu labeled "----Select the collection to insert into the page----"; "Display Function:" with an empty text box; and another dropdown menu labeled "-----Select the display function or fill it in above----". On the right side, there are three buttons: "Insert", "Cancel", and "Help".

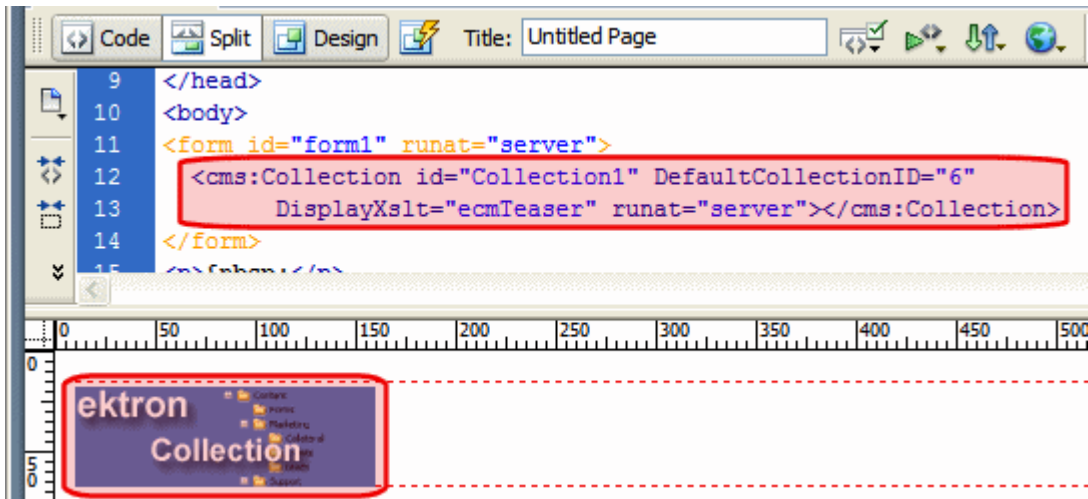
4. Use the following table to assist you with inserting a Collection.

Field	Description
Control ID <hr/> Used in ASP.NET only. <hr/>	A unique name for the function you are inserting.
Collection ID	If you know the collection ID, enter it into the textbox. If you do not know the Collection ID, you can navigate to it using the Select Collection drop down box.
Select Collection	The drop-down list is filled with a list of all the Collection titles that you have privileges to. When a title is selected, the Collection ID textbox is filled with that Collection's id number. If you insert the Collection ID, you do not need to select a category.
Display Type	Determines how the information on the page is displayed. None - databind only ecmNavigation - lists the title of every content block in the collection See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists the title of every content block in the collection plus the content summary See Also: "Example of ecmTeaser Display" on page 85
LinkTarget <hr/> Used in ASP.NET only. <hr/>	Defines the way a link acts when a link is clicked. Choices are: _blank - This target causes the link to always be loaded in a new blank window. This window is not named. _self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. _parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. _top - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.
Dynamic Content Parameter <hr/> Used in ASP.NET only. <hr/>	Checks the QueryString for this value and replaces the collection with a content block when specified. Leave blank to always display the collection.

Field	Description
Dynamic Collection Parameter <hr/> Used in ASP.NET only.	To make this collection dynamic, select coll_id . When you do, the function uses the collection passed as a URL parameter.
Random Output <hr/> Used in ASP.NET only.	Displays random content within a collection. None - Displays all content in collection. Random Teaser - Displays a random content block's teaser. Random Body - Displays a random content block's body.
# of seconds - CacheInterval <hr/> Used in ASP.NET only.	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36 .
Hide <hr/> Used in ASP.NET only.	Used to hide a collection in design time and run time. <input checked="" type="checkbox"/> Checked = Hide collection <input type="checkbox"/> Unchecked = Show collection <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Visible <hr/> Used in ASP.NET only.	Used to show or hide the function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only.	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"
Wrap Tag <hr/> Used in ASP.NET only.	Allows a developer to specify a server control's tag. The default is Span . Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element. Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.

- When completed, click **insert**.

A graphic defines where the collection appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:Collection id="Collection1" DefaultCollectionID="6" DisplayXslt="ecmTeaser" runat="server"></cms:Collection></code>
ASP	<code><% ecmCollection 6,"ecmNavigation" %></code>
ColdFusion	<code><cfmodule id="6" displayFunction="ecmNavigation" template="#request.ecm.AppPath#ecmCollection.cfm"></code>
JSP	<code><%= ecmCollection(6,"ecmTeaser") %></code>
PHP	<code><?php ecmCollection(6,"ecmNavigation"); ?></code>

Content Block

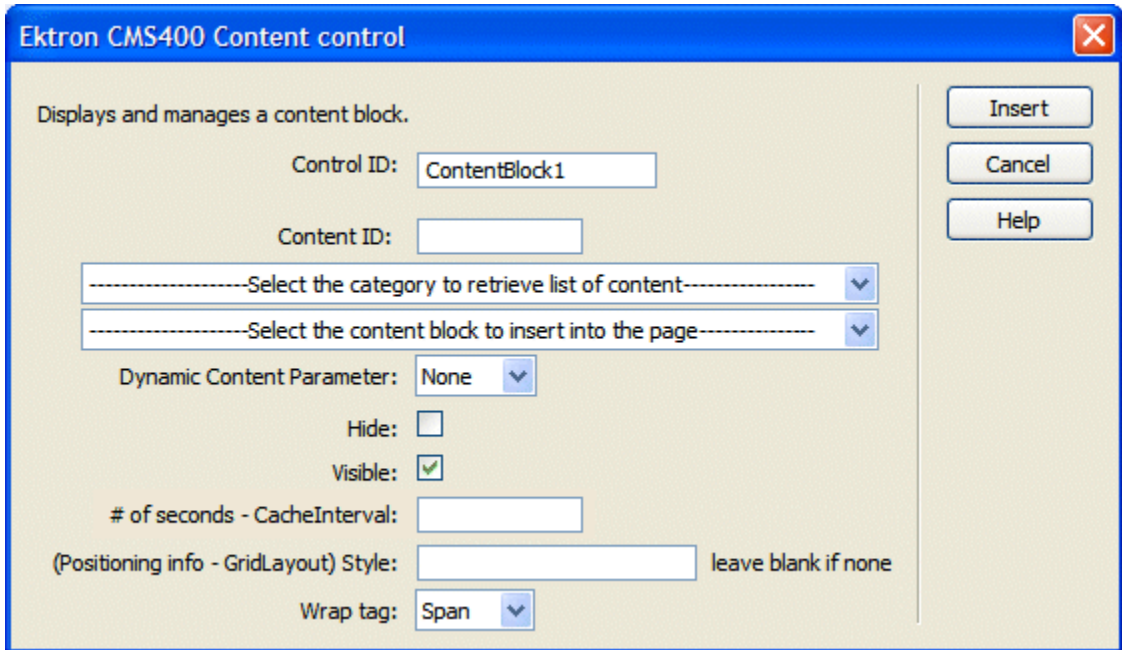
This tag appears on the following menu(s):

- Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

For a description of the Content Block function, see ["ContentBlock Server Control" on page 91](#).

To insert a content block:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Content**.
3. The Insert Content box is displayed



Ektron CMS400 Content control

Displays and manages a content block.

Control ID:

Content ID:

-----Select the category to retrieve list of content----- ▾

-----Select the content block to insert into the page----- ▾

Dynamic Content Parameter: ▾

Hide: ☐

Visible: ☒

of seconds - CacheInterval:

(Positioning info - GridLayout) Style: leave blank if none

Wrap tag: ▾

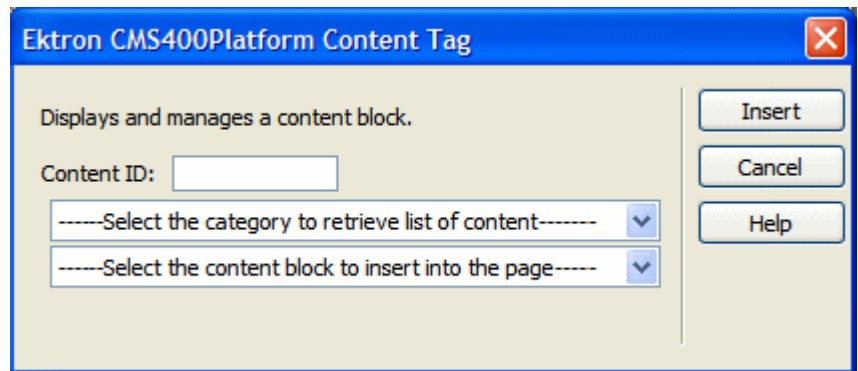
Insert

Cancel

Help

IMPORTANT!

When inserting a Content Block using ASP, CF, JSP or PHP, a different dialog box appears.



Ektron CMS400Platform Content Tag

Displays and manages a content block.

Content ID:

-----Select the category to retrieve list of content----- ▾

-----Select the content block to insert into the page----- ▾

Insert

Cancel

Help

4. Use the following table to help you insert a content block.

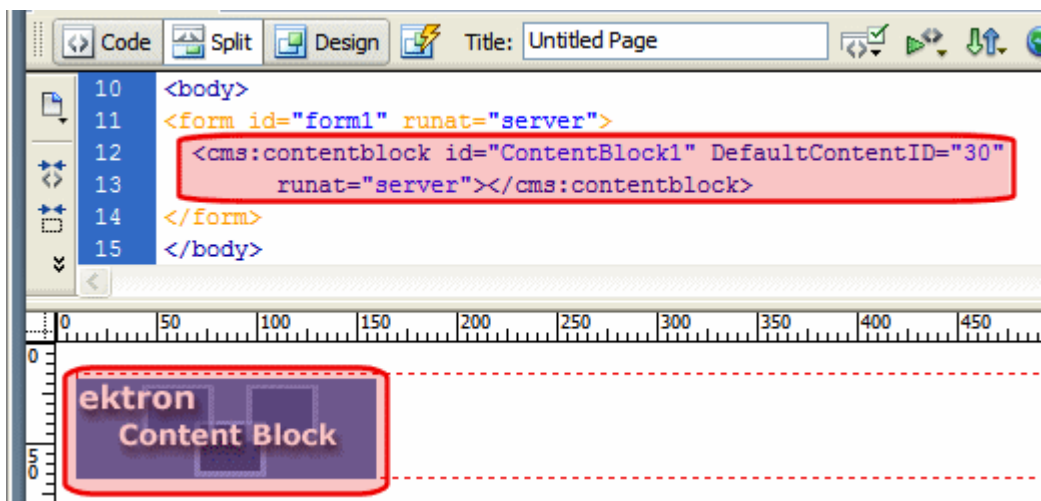
Field	Description
Control ID <hr/> Used in ASP.NET only. <hr/>	A unique name for the function you are inserting.
Content ID	Specify the ID number of the content block that you wish to use as a content block on the Web page. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content drop down boxes.
Select Category	Choose the folder where the content is located. If you insert the content ID, you do not need to select a category.
Select Content	Select the content block to use from the list. If you insert the content ID, you do not need to select content.
Dynamic Content Parameter <hr/> Used in ASP.NET only. <hr/>	<p>Determines if content is displayed as static or passed dynamically. By making content dynamic, when the content displayed on a Web page contains a link to another piece of content, you can display the second piece of content in the same Web page space.</p> <p>None: Displays a static content block.</p> <p>id: Content block is passed as a URL parameter.</p> <hr/> <p>Important: If you are using an ASP, CF, JSP or PHP page, you must use the Dynamic Content Block Tag to create a dynamic content block. See "Dynamic Content Block" on page 333</p> <hr/>
Hide <hr/> Used in ASP.NET only. <hr/>	<p>Used to hide a content block in design time and run time.</p> <p>True = Hide content block</p> <p>False = Show content block</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible <hr/> Used in ASP.NET only. <hr/>	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>

Field	Description
# of seconds - CacheInterval <hr/> Used in ASP.NET only. <hr/>	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only. <hr/>	Set position information of a control as absolute, when using a GridLayout Web page. For example: <code>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</code>
Wrap Tag <hr/> Used in ASP.NET only. <hr/>	Allows a developer to specify a server control's tag. The default is Span . Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element. Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.

NOTE You see only the list of Ektron CMS400.NET content blocks if you are working online.

5. Click **insert**.

A graphic indicates the static content block's location on the template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<cms:contentblock id="ContentBlock1" DefaultContentID="12" runat="server"></cms:contentblock>
ASP	<% ecmContentBlock(1) %>
ColdFusion	<cfmodule id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm">
JSP	<%= ecmContentBlock(12) %>
PHP	<?php ecmContentBlock(1); ?>

Content List

For a description of the Content List function, see ["ContentList Server Control" on page 100](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert the Content List function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Content List**.

3. The Insert Ektron CMS400.NET Content List box appears.

Ektron CMS400 Content List

ContentList is used to display a list of content block

Control ID:

ContentIds

ContentList control can programatically read data out of Content Block vs setting the ContentIds

MetaTag - of Type Content Selector

DefaultContentID

▼

▼

Dynamic Content Parameter ▼

Order By: ▼ ▼

Display Type ▼ LinkTarget ▼

Random Output: ▼

of seconds - CacheInterval:

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style: leave blank if none

Wrap tag: ▼

Insert Cancel Help

4. Use the following table to assist you with inserting the Content List function.

Field	Description
Control ID	A unique name for the function you are inserting.
ContentIds	A comma delimited list of content blocks IDs. <i>See Also: "Using the ContentID Property to Display a Content List" on page 109</i>

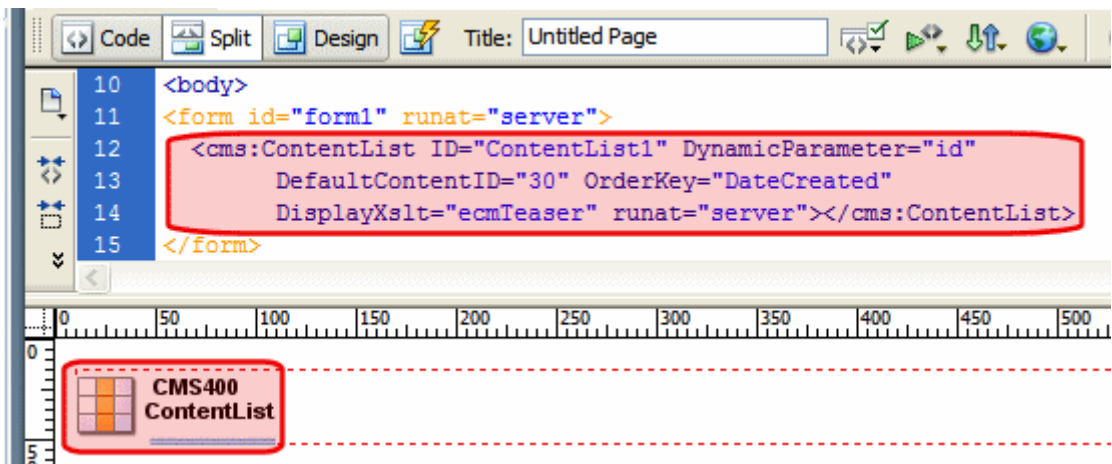
Field	Description
MetaTag	<p>Specify a Metadata definition whose type is Content Selector. When you do, the associated list of content items will appear where you place the server control.</p> <hr/> <p>Warning!: You cannot insert other metadata types.</p> <hr/> <p>This works with the DefaultContentID property. See Also: "Using the DefaultContentID and MetaTag Properties" on page 105</p> <p>For more information about using metadata to assign a list of related content to a content item, see the Ektron CMS400.NET Administrator manual section "Managing Content" > "Metadata" > "Types of Metadata" > "Related Content Metadata."</p>
DefaultContentID	<p>Set content id value. Once set, content IDs are generated from the MetaTag value for this content. If you do not know the content's ID, you can navigate to it using the Select Category and Select Content drop down boxes.</p>
DynamicParameter	<p>To make the content dynamic, select id. When you do, this server control uses the content passed as a URL parameter.</p>
Order By	<p>Sort the list by one of the values below and choose whether the list is sorted in Ascending or Descending order.</p> <ul style="list-style-type: none"> • Title - the order of the content's title. • Date Modified - the last date the content was modified. • Date Created - the date the content was created. • Last Author First Name - the last editor's first name. • Last Author Last Name - the last editor's last name. • OrderOfTheIds - preserves the content ID order based on the list in the ContentIds property.
Display Type	<p>Determines how information on the page is displayed.</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of each content block See Also: "Example of ecmNavigation Display" on page 85</p> <p>ecmTeaser - lists the title of each content block plus the content summary See Also: "Example of ecmTeaser Display" on page 85</p> <p>ecmUnOrderedList - sorts the list in no particular order. Shows the title and content summary</p>

Field	Description
LinkTarget	<p>Defines the way a link acts when a link is clicked.</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>
Random Output	Set to True if you want to randomly display one content block in the specified folder. The content changes each time a user views the page.
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Content List icon appears and indicates the Content List's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<pre><cms:ContentList ID="ContentList1" DynamicParameter="id" DefaultContentID="30" OrderKey="DateCreated" DisplayXslt="ecmTeaser" runat="server"></cms:ContentList></pre>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Content Rating

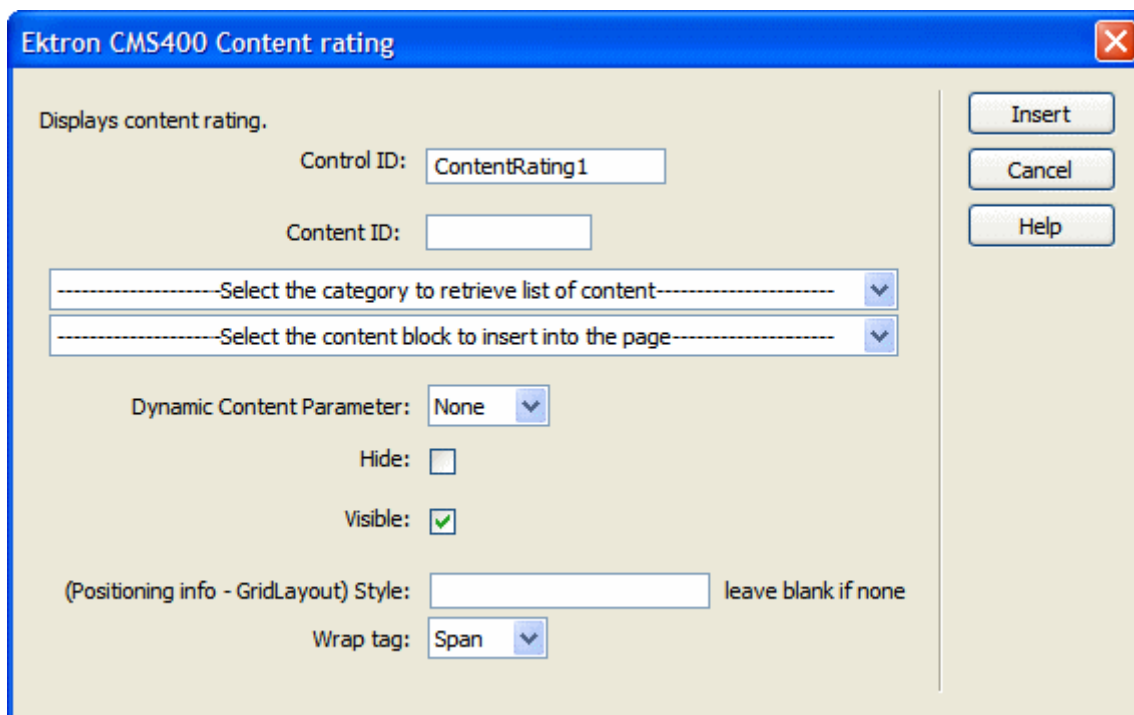
For a description of the Content Rating function, see "[ContentRating Server Control](#)" on page 110.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert the Content Rating function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **ContentRating**.
3. The Insert Ektron CMS400.NET Content Rating box appears.



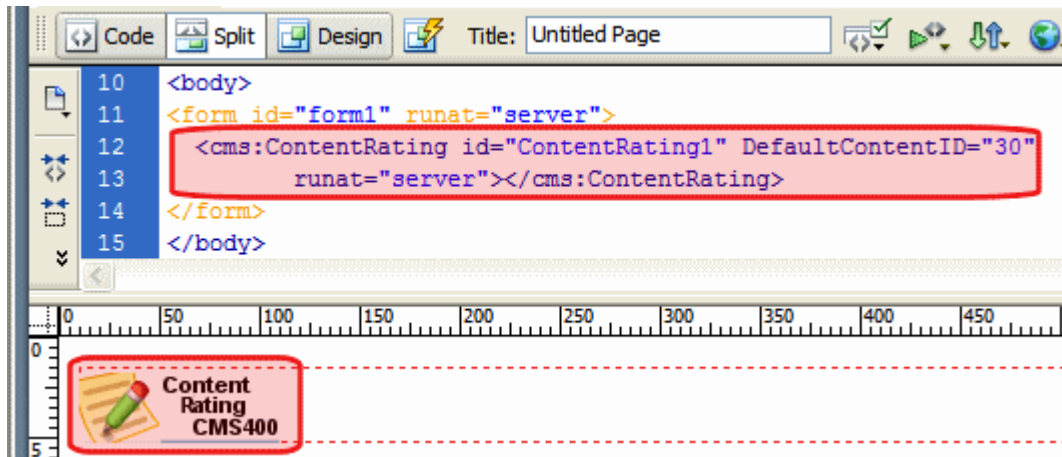
4. Use the following table to assist you with inserting the Content Rating function.

Field	Description
Control ID	A unique name for the function you are inserting.
Content ID	Specify the ID number of a content item being tracked by this function. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content drop down boxes.

Field	Description
Dynamic Content Parameter	To make this content rating control dynamic, select id . When you do, this server control is attached to the content block passed as a URL parameter.
Hide	<p>Used to hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

An Content Rating icon appears and indicates the Content Rating's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:ContentRating id="ContentRating1" DefaultContentID="30" runat="server"></cms:ContentRating></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Content XSLT Tag...

The insert content XSLT command allows you to insert an XML content block, and specify an XSLT (other than the default) to be applied to it.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > XML Function...**

To insert a content XSLT tag, perform the following steps:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET XML functions, click on **Insert Content XSLT Tag...**

3. The Insert Ektron CMS400.NET Content and Apply an XSLT Tag box is displayed.

Insert Ektron CMS400 Content and apply an XSLT

Displays and manages a content block with an XSLT.

Content ID:

-----Select the category to retrieve list of content-----

-----Select the content block to insert into the page-----

-----Select how to pick the XSLT -----

The XSLT must have a fully qualified path like: C:\somepath\myxslt.xml

XSLT:

Insert

Cancel

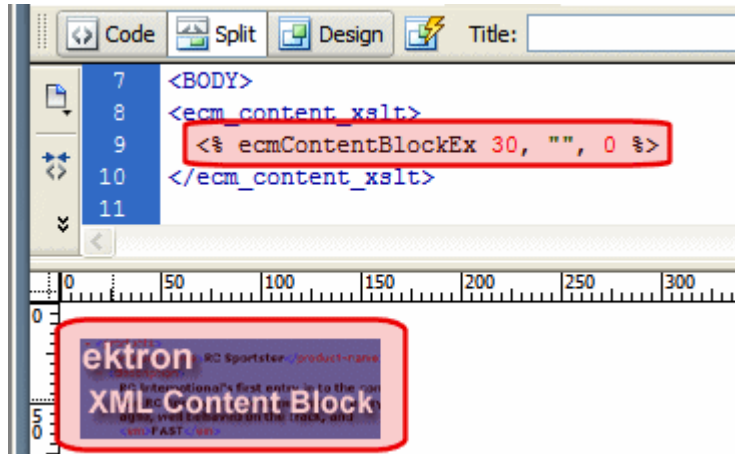
Help

4. Use the following table to assist you with this form.

Field	Description
Content ID	Either type in, or select from the drop down lists, the ID number of the XML content to insert.
Select XSLT	From the drop down list, select the XSLT you would like to apply to the XML content block.
Enter XSLT	If "Enter XSLT" is selected, you can insert the path and file name of the XSLT to apply to the content.

5. Click **insert**.

A graphic showing you where the XML content block will be placed on the template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
.NET	This function is not available as an Ektron Dreamweaver Extension for .NET.
ASP	<code><% ecmContentBlockEx 30, "", 1 %></code>
ColdFusion	<code><cfmodule id="5" xsltpath="" xslttype="1" template="#request.ecm.AppPath#ecmContentBlockEx.cfm"></code>
JSP	<code><%= ecmContentBlockEx(24, "", 0) %></code>
PHP	<code><?php ecmContentBlockEx (5, "", 1); ?></code>

DHTML Menu

For a description of the DHTML Menu function, see ["DhtmlMenu Server Control" on page 160](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Control...**

To insert a DHTML Menu:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **DHTML Menu**
The DHTML Menu Control box appears.

Ektron CMS400 DhtmlMenu Control

DhtmlMenu

Control ID:

DefaultMenuID:

-----Select the from a list of menus-----

Align : HorizontalDisplay: ☐

CssClass:

MenuHoverOverColor : SubMenuBackColor:

SubMenuHoverOverColor : BackColor

ForeColor : Font-Bold ☐ Font_Italic ☐

of seconds - CacheInterval::

FontNames : Font-Size

SubMenuItemHeight : SubMenuItemWidth

SubMenuTopOffset : MenuStyle:

Stylesheet:

Dynamic menu Parameter:

Wrap tag : Visible: ☒ Hide ☐

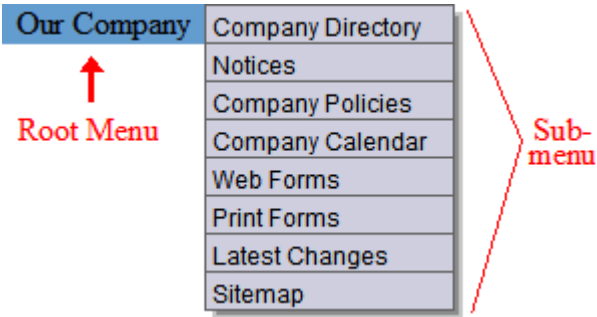
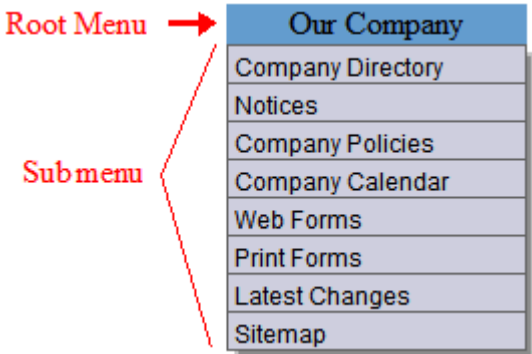
(Positioning info - GridLayout) Style: leave blank if none

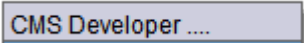

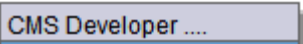

Insert Cancel Help

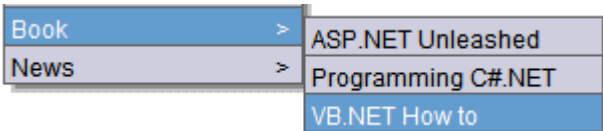
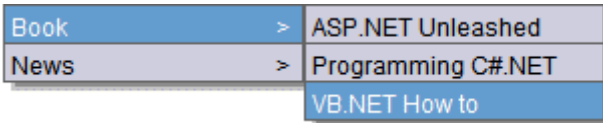
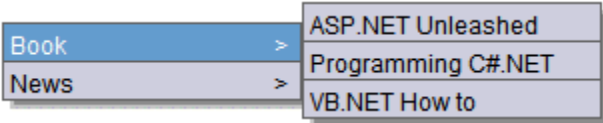

3. Use the following table to assist you with inserting a DHTML Menu.

Field	Description
Control ID	A unique name for the function you are inserting.

Field	Description
DefaultMenuID	The numeric id of the menu that you want to insert. If you do not know the DefaultMenuID path, you can navigate to it using the Select Menu drop down box.
Select Menu	Select a menu to be used. When a menu is selected, the DefaultMenuID textbox is filled with that menu's id number. If you insert the DefaultMenuID, you do not need to select a menu.
Align	Specifies the horizontal alignment of the root menu text. The default is Center . NotSet - Use the default setting. Left - Aligns text to the left. Center - Aligns text to the center. Right - Aligns text to the right. Justify - Aligns text justified to both margins.

Field	Description				
HorizontalDisplay	<p>Displays submenu to the right of the root menu. This only affects the first submenu of the root menu. All subsequent submenus appear to the right of each submenu.</p> <p>The default is False</p> <p>True = Submenu appears to the right of the root menu</p>  <p>False = Submenu appears below the root menu</p> 				
CssClass	<p>Apply a .css class to the root menu. Enter the class name from the .css defined in the Style sheet property.</p> <p>Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet property.</p> <p>For example, if you enter lowtext in the property, the root menu changes as shown below.</p> <table border="1"> <tr> <td>No CssClass applied</td><td>lowText CssClass applied</td></tr> <tr> <td>MenuExample</td><td>MenuExample</td></tr> </table>	No CssClass applied	lowText CssClass applied	MenuExample	MenuExample
No CssClass applied	lowText CssClass applied				
MenuExample	MenuExample				

Field	Description
MenuHoverOverColor	Sets the background color for the root menu when hovering over it.
SubMenuBackColor	Sets the background color for the submenu.
SubMenuHoverOverColor	Sets the background color for the submenu when hovering over it.
BackColor	Sets the background color for the root menu. This property appears in the Appearance category
ForeColor	Sets the color of the font for the root menu. This property appears in the Appearance category.
Font-Bold	Checking this box allows the font to appear bold .
Font_Italic	Checking this box allows the font to appear <i>italics</i> .
FontNames	Enter the name of a font to use.
Font-Size	The size of the font.
SubMenuItemHeight	<p>Sets the height of each submenu item in pixels. The Default is 20</p> <p>Example set to 20: </p> <p>Example set to 30: </p>
SubMenuItemWidth	<p>Sets the width of each submenu item in pixels. The Default is 150</p> <p>Example set to 150: </p> <p>Example set to 130: </p>

Field	Description
SubMenuTopOffset	<p>Sets the location of the top of the submenu. This number is in pixels. The default is 5</p> <p>Example set to 5:</p>  <p>Example set to 0 (zero):</p>  <p>Example set to -10:</p> 
MenuStyle	<p>A specified inline style that defines the root menu.</p> <p>For example, if you enter the following in the MenuStyle property box:</p> <pre>font: bold 18px Arial;color:royalblue; background-color:slateblue</pre> 
StyleSheet	<p>Specify the location and style sheet for the submenu. For example: \Workarea\csslib\mytest.css.</p> <p>Leave blank to use the default.</p> <p>pop_style.css is the default style sheet. It is located in \webroot\CMS400demo\Workarea\csslib. For more information on style sheets, see "Using a Style Sheet" on page 165</p> <p>The Stylesheet property can be used in conjunction with the CssClass property to define the root menu.</p>
Dynamic Menu Parameter	<p>Checks the QueryString for the menu value and replaces the menu with content when ID is specified. Leave blank to always display the menu.</p> <p>None - Use Default</p> <p>menu_ID - The control uses the menu passed as a URL parameter.</p>

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Hide	<p>Used to hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

4. When completed, click **insert**.

A graphic defines where the DHTML Menu appears on your template.

Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:DhtmlMenu id="DhtmlMenu1" DefaultMenuID="16" ForeColor="" runat="server"></cms:DhtmlMenu></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.

Platform	Code View
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Discussion Forum

For a description of the Discussion Forum function, see ["Forum Server Control" on page 120](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert the Discussion Forum function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Discussion Forum**.
3. The Insert Ektron CMS400.NET Discussion Forum box appears.

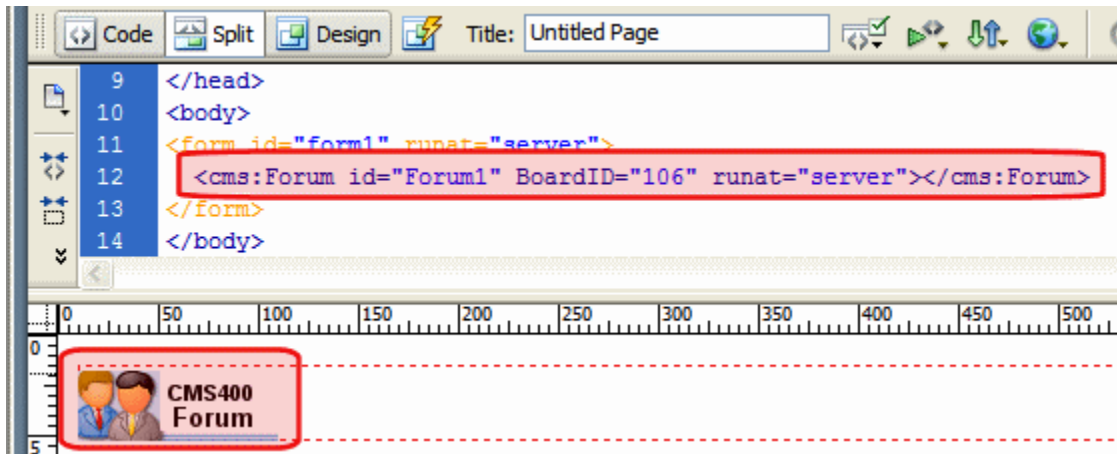
4. Use the following table to assist you with inserting the Blog function.

Field	Description
Control ID	A unique name for the function you are inserting.
Board ID	The ID of the Discussion Board in CMS400.NET. If you do not know the Board ID, you can choose it from the Select Folder drop down box.

Field	Description
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide	<p>Used to hide a function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.</p>

5. Click **insert**.

A Discussion Forum icon appears and indicates the Discussion Forum's location on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:Forum id="Forum1" BoardID="106" runat="server"></cms:Forum></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Display By Meta Value

For a description of the Display By Meta Value function, see ["MetaDataList Server Control" on page 171](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert Display By Meta Value:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Display By Meta Value**
The Display By Meta Value box appears.

3. Use the following table to assist you with inserting Display By Meta Value.

Field	Description
Control ID	A unique name for the function you are inserting.
FolderID	The numeric id of the folder that you want to use. If you do not know the folder ID, you can navigate to it using the Select folder drop down box.

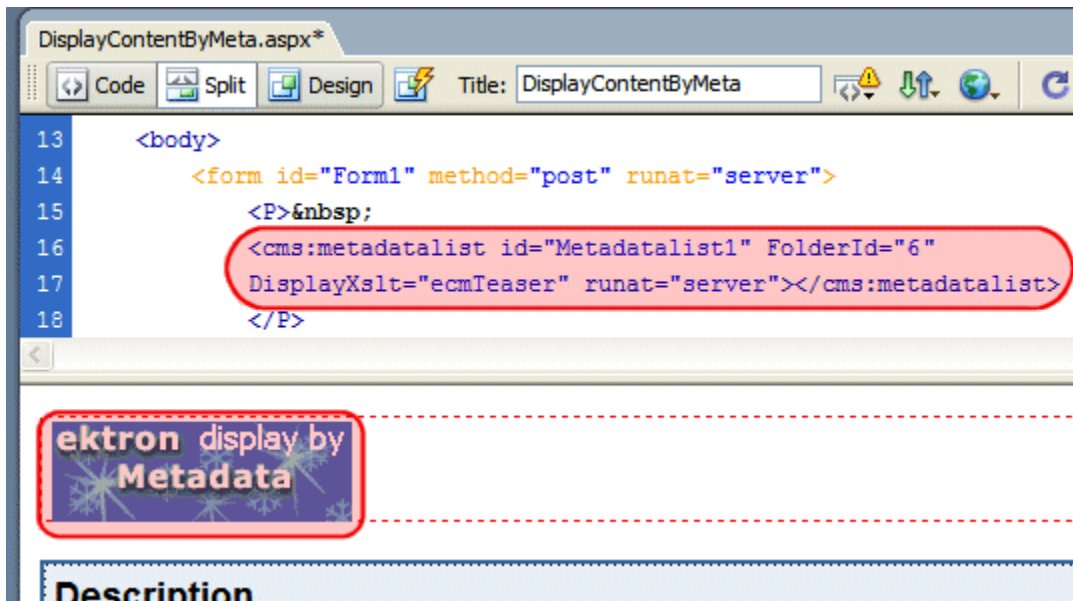
Field	Description
Select Folder	Select a folder path to be used. When a folder is selected, the folderID textbox is filled with that folder's id number. If you insert the Folder ID, you do not need to select a folder.
Recursive	Determines whether the Display By Meta Value control uses this folder's child folders.
Order By	<ul style="list-style-type: none"> Specify how you want to order the results that are generated. Also specify if you want the results in ascending or descending order. Title - The title of the content block Date Created - The date the content block was created Date Modified - The date the content block was last modified Last Author Last Name - The last author's last name Last Author First Name - The last author's first name ID - The content blocks ID Ascending - Alphabetic or numerical order Descending - Reverse alphabetic or numerical order
Max Number of results	The Maximum number of content blocks returned (0=unlimited).
ExactPhrase	<p>Determines whether the KeyWordValue needs to match the phrase exactly. For example, if "site" is the KeyWordValue and the title of a content block is "Welcome to the site" and ExactPhrase is set to true, you would not see the content block in the metadata list. This is because "site" does not equal "Welcome to the site".</p> <p><input checked="" type="checkbox"/> Checked = Match the exact phrase</p> <p><input type="checkbox"/> Unchecked = Doesn't need to match exact phrase</p>

Field	Description
LinkTarget	<p>Defines the way a link acts when a link is clicked.</p> <p>Choices are:</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>
KeywordName	<p>The KeywordName is the container where the KeywordValues are located. Some examples of a KeywordName might be “Keywords or Title.”</p> <p>For information on creating KeywordNames for Metadata, see the Ektron CMS400.NET Administrators Manual > “Managing Content” > “Metadata” > “Managing Metadata Definitions” > “Adding a Metadata Definitions”</p>
KeywordValue	<p>The value that is associated with the KeywordName.</p> <p>Some examples of a KeywordValue might be “Home; page; or company”</p>
Display Type	<p>Determines how the information on the page is displayed.</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder</p> <p>See Also: "Example of ecmNavigation Display" on page 85</p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary</p> <p>See Also: "Example of ecmTeaser Display" on page 85</p>
Dynamic Content Parameter	<p>Checks the QueryString for the content block ID value and replaces the Display By Meta Value Control with content when an ID is specified. Leave blank to always display the Display By Meta Value Control.</p> <p>None - Use Default</p> <p>ID - The control uses the content block passed as a URL parameter.</p>
GetHtml	<p>Set to True if you want to retrieve and display content (html body) for all content blocks in the collection. For example, display content inside a web server control such as a GridView.</p>

Field	Description
Content Type	<p>Defines what type of list summary appears.</p> <p>Content - Displays only content blocks in the folder.</p> <p>All Types - Displays all content in the folder.</p> <p>Forms - Displays only forms in the folder.</p> <p>Archive_Content - Displays archived content from the folder.</p> <p>Archive_Forms - Displays archived forms from the folder.</p>
Hide	<p>Used to hide the function in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page. For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

4. When completed, click **insert**.

A graphic defines where the Display By Meta Value appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:metadatalist id="Metadatalist1" FolderId="6" DisplayXslt="ecmTeaser" runat="server"></cms:metadatalist></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Dynamic Content Block

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

For a description of a dynamic content block, see ["Dynamic Content Block" on page 419](#)

To insert a dynamic content block tag:

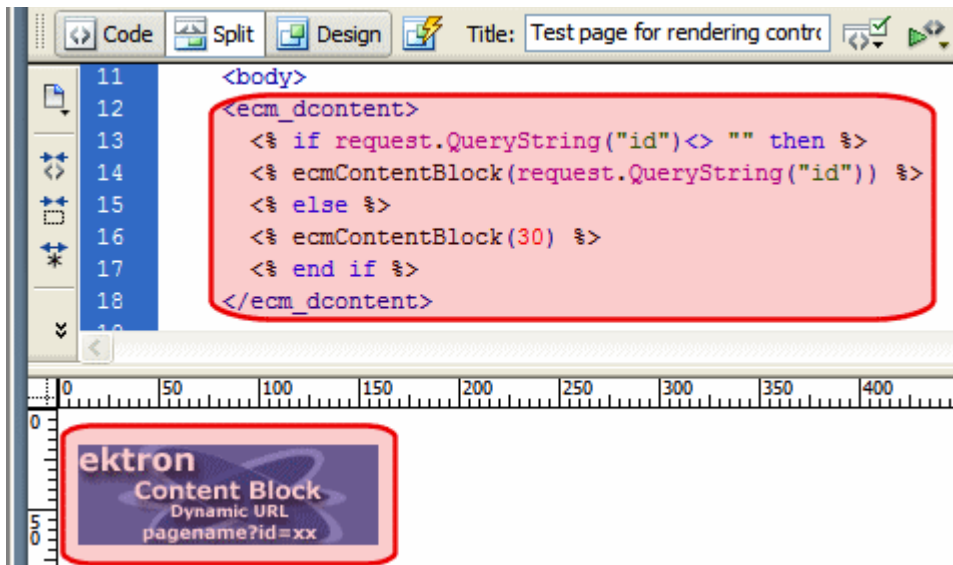
1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Dynamic Tag...**
3. The Insert Ektron Dynamic Tag box is displayed

4. Use the following table to assist you with inserting a dynamic content block function.

Field	Description
Insert Default Content	If you want a static content block to be displayed if no ID is passed through the URL, place a check in this box.
Default Content ID	Specify the ID number of the content block that you wish to use as a default content block on the Web page.
Select Content	If you are working online, you may select the content block to use as a default content block from the list.

- When completed, click **Insert**.

A graphic to define where the dynamic content block has been placed on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
.NET	For ASP.NET pages, use the CMS400.NET (Server Control) Content Block tag and set the dynamic parameter to id. See "Content Block" on page 306
ASP	<pre> <% if request.QueryString("id")<> "" then ecmContentBlock(request.QueryString("id")) else ecmContentBlock(1) end if %> </pre>
ColdFusion	<pre> <cfif isdefined("url.id")> <cfmodule id="#url.id#" template="#request.ecm.AppPath#ecmContentBlock.cfm"> <cfelse> <cfmodule id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm"> </cfif> </pre>
JSP	Inserting a Dynamic Content Block using Ektron's Dreamweaver Extensions is not supported in JSP.
PHP	<pre> <?php \$id =GetURL_Id(1); ecmContentBlock(\$id); ?> </pre>

Folder Bread Crumb

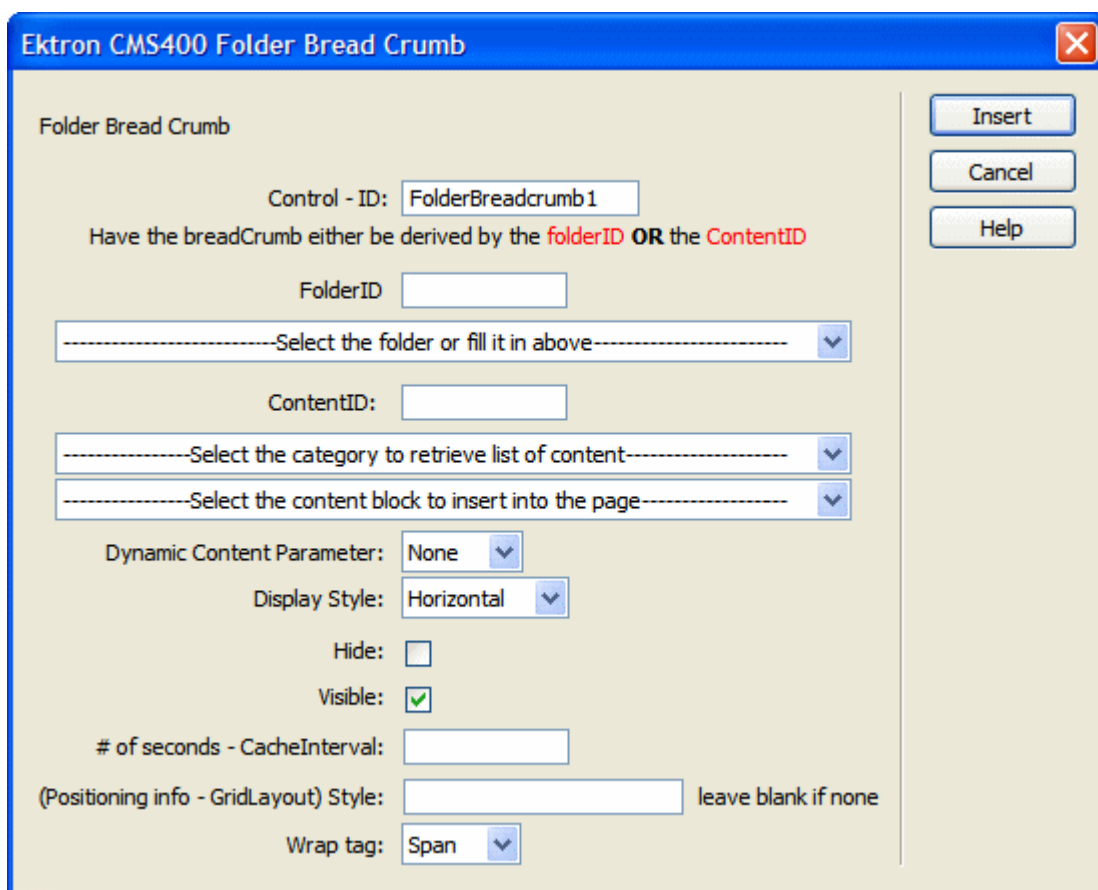
For a description of the Folder Bread Crumb function, see "[FolderBreadcrumb Server Control](#)" on page 69.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Controls...**

To insert a Folder Bread Crumb Trail:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Folder BreadCrumb**.
3. The Folder Bread Crumb dialog box appears.



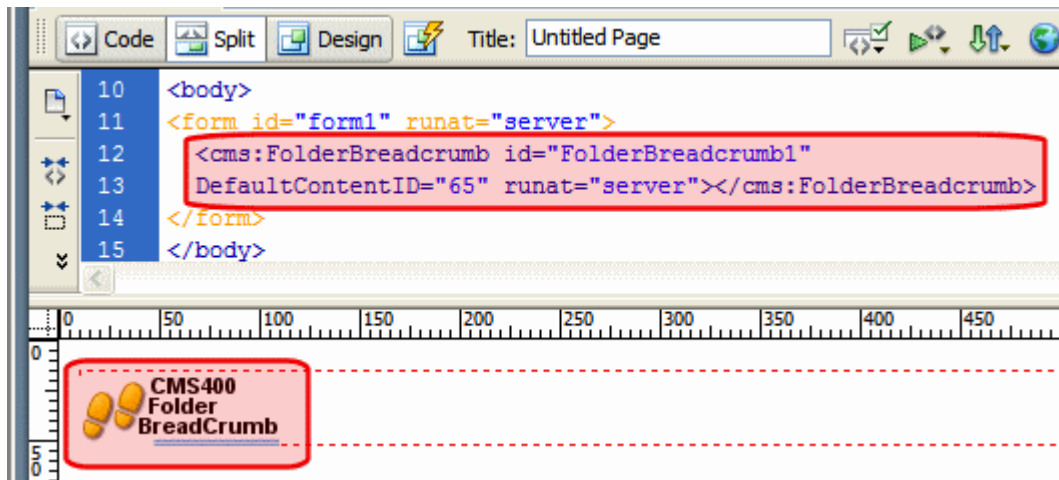
4. Use the following table to assist you with inserting a Folder Bread Crumb.

Field	Description
Control ID	A unique name for the function you are inserting.

Field	Description
FolderID	The numeric id of the folder that you want to use. If you do not know the folder ID, you can navigate to it using the Select folder drop down box.
Content ID	Gets or Sets the ContentID. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content drop down boxes.
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.
DisplayStyle	Indicate how to display the folder bread crumb: horizontally or vertically. The default is Horizontal.
Hide	<p>Used to hide a Language API in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide</p> <p><input type="checkbox"/> Unchecked = Show Language API</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). See Also: " Caching with Server Controls " on page 36.
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **Insert**.

A graphic defines where the Folder Bread Crumb appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<pre> <cms:FolderBreadcrumb id="FolderBreadcrumb1" DefaultContentID="65" runat="server"></cms:FolderBreadcrumb> </pre>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Form

For a description of the Form function, see ["FormBlock Server Control" on page 117](#)

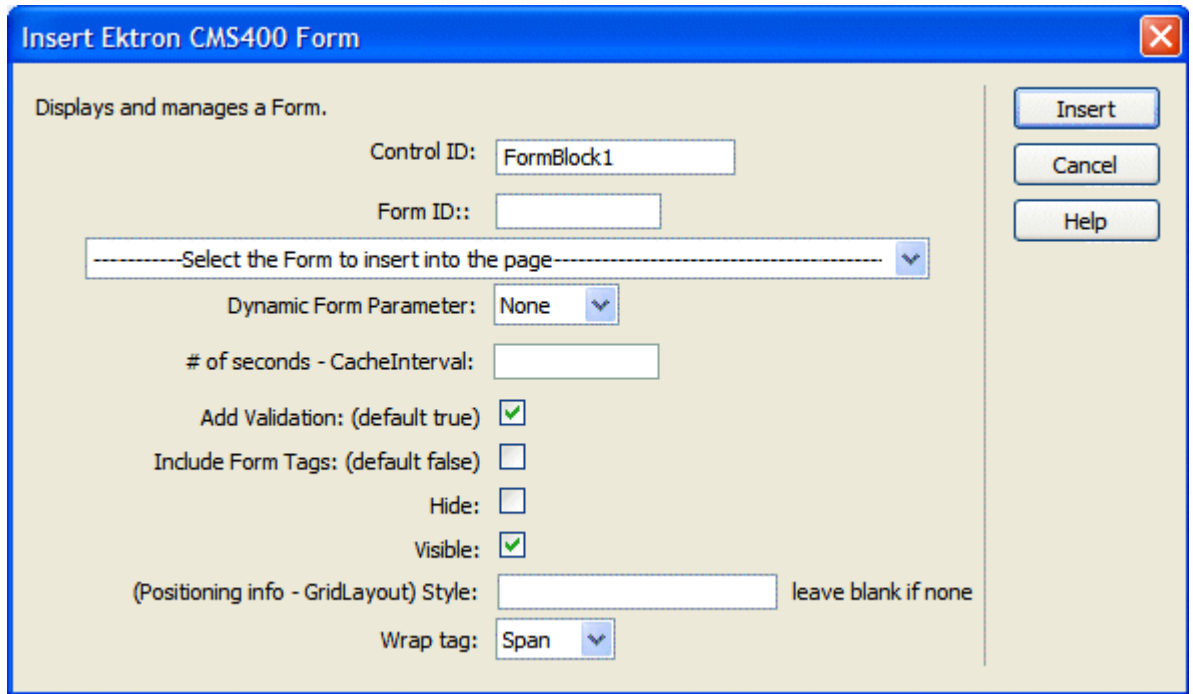
This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a form:

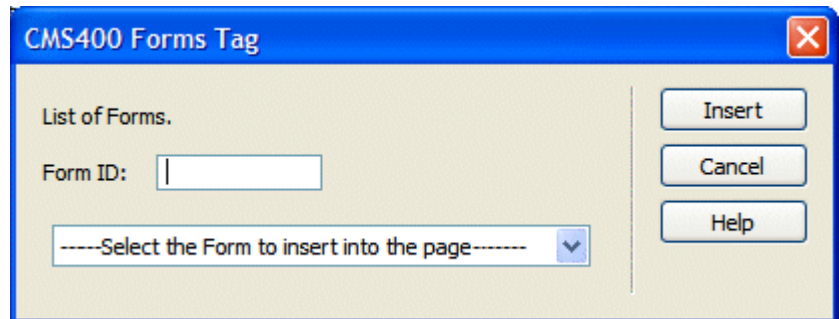
- Move the blinking cursor to the area that you want to place it.
- From the list of Ektron CMS400.NET Content functions, click **Form**

3. The Insert Form box is displayed.



IMPORTANT!

When inserting a form tag using ASP, CF, JSP or PHP, a different dialog box appears.



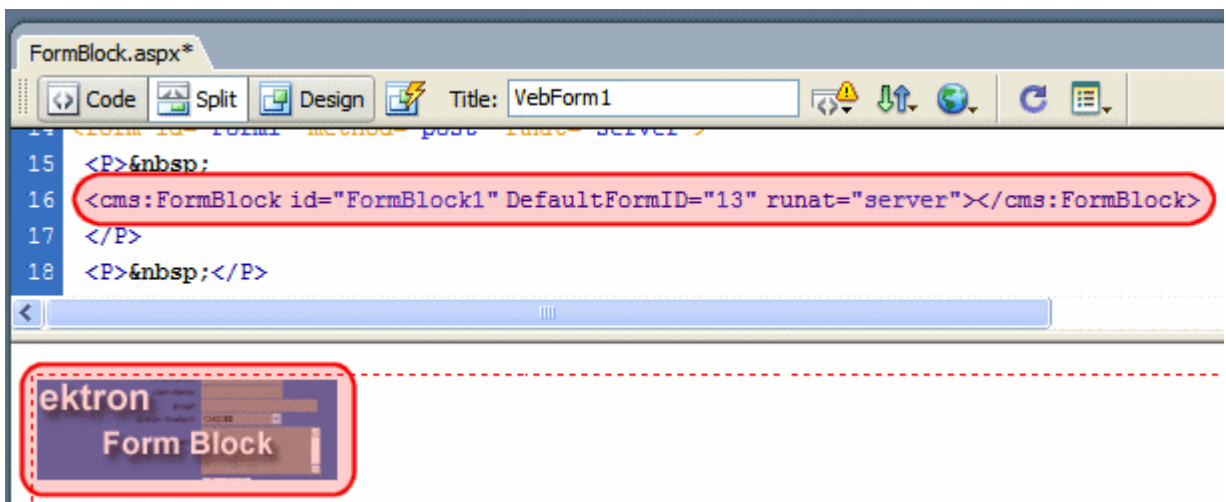
4. Use the following table to assist you with inserting a form.

Field	Description
Control ID	A unique name for the function you are inserting.
<u>Used in ASP.NET only.</u>	

Field	Description
Form ID	Type in the form ID that you want to insert. If you do not know the Form ID, you can navigate to it using the Select Form drop down box.
Select Form	The drop-down box is populated with a list of all the form titles. When a title is selected, the form ID textbox is filled with that form's id number. If you insert the Form ID, you do not need to select a form.
Dynamic Form Parameter <u>Used in ASP.NET only.</u>	To make this form block dynamic, select ekfrm . When you do, this control uses the form block passed as a URL parameter.
# of seconds - CacheInterval <u>Used in ASP.NET only.</u>	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Add Validation <u>Used in ASP.NET only.</u>	When checked , adds Validation to your main runat=server form.
Include Form Tags <u>Used in ASP.NET only.</u>	Determines if tags are generated automatically or manually. See Also: "Automatic versus Manual Generation of Form Tags" on page 119
Hide <u>Used in ASP.NET only.</u>	Used to hide a form in design time and run time. <input checked="" type="checkbox"/> Checked = Hide form <input type="checkbox"/> Unchecked = Show form <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Visible <u>Used in ASP.NET only.</u>	Used to show or hide the function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden.

Field	Description
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only.	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"
Wrap Tag <hr/> Used in ASP.NET only.	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code.

- When completed, click **Insert**.
A graphic defines where the form appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:FormBlock id="FormBlock1" DefaultFormID="13" runat="server"></cms:FormBlock></code>
ASP	<code><% ecmFormBlock (1) %></code>
ColdFusion	<code><cfmodule id="1" template="#request.ecm.AppPath#ecmFormBlock.cfm"></code>

Platform	Code View
JSP	<code><%= ecmFormBlock(1) %></code>
PHP	<code><?php ecmFormBlock (1); ?></code>

Language API

For a description of the Language API function, see "[LanguageAPI Server Control](#)" on page 130.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a Language API:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Language API**
3. The Language API box appears.

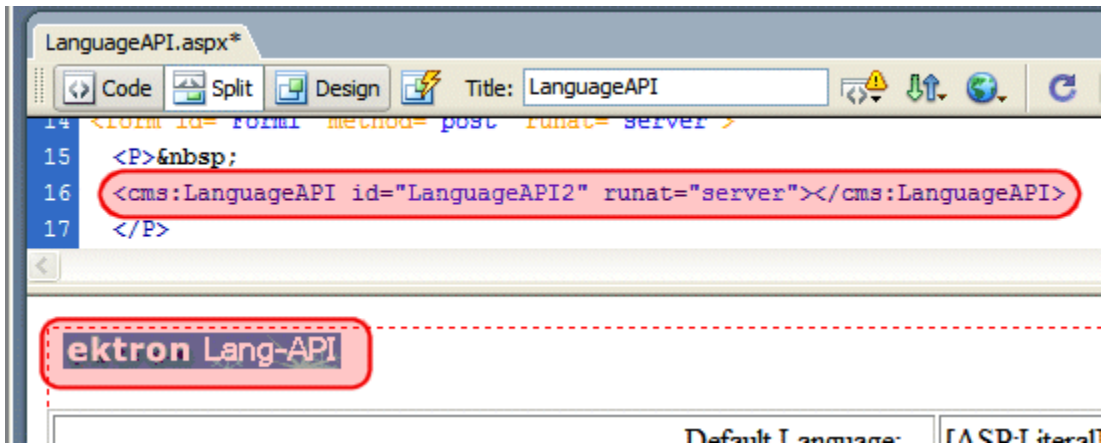
4. Use the following table to assist you with inserting a Language API box.

Field	Description
Control ID	A unique name for the function you are inserting.

Field	Description
Hide	<p>Used to hide a Language API in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide Language API</p> <p><input type="checkbox"/> Unchecked = Show Language API</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

5. When completed, click **insert**.

A graphic defines where the Language API appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:LanguageAPI id="LanguageAPI2" runat="server"></cms:LanguageAPI></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Language Select Box

For a description of the Language Select Box, see ["LanguageSelect Server Control" on page 133](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a Language Select Box:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Language Select**

3. The Language Select box appears.

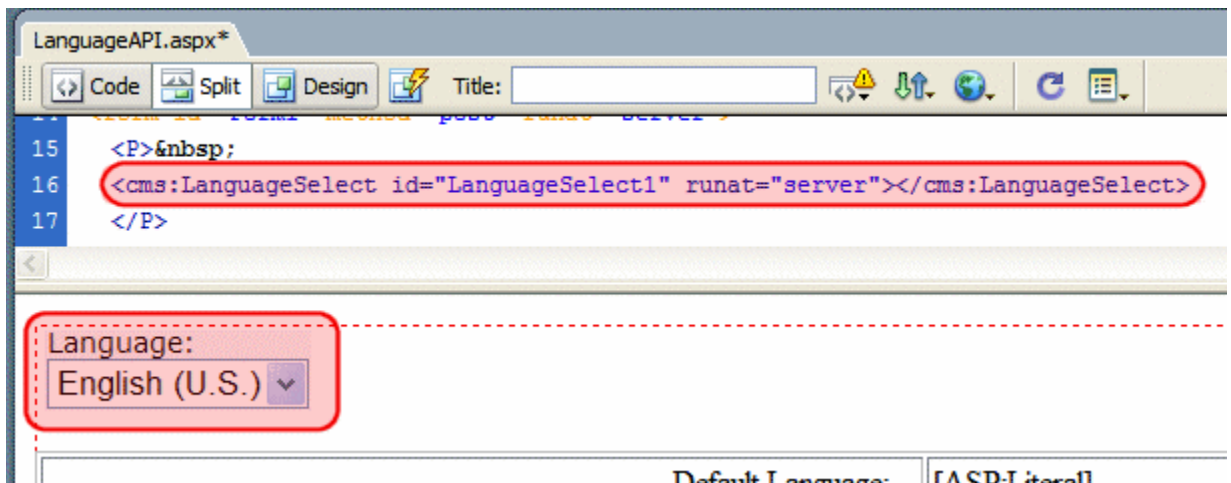
4. Use the following table to assist you with inserting a Language Select box.

Field	Description
Control ID	A unique name for the function you are inserting.
Hide	<p>Used to hide a function in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **insert**.

A graphic defines where the Language Select appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:LanguageSelect id="LanguageSelect1" runat="server"></cms:LanguageSelect></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

List Summary

For a description of the List Summary function, see "[ListSummary Server Control](#)" on page 135.

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a list summary:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **List Summary**.
3. The Insert Ektron List Summary box is displayed.

Ektron CMS400 List Summary Control

ListSummary is used to display a list of content block from a folder.

Control ID:

FolderID:

-----Select the folder or fill it in above-----

Recursive: ☐

Order By:

Max Number of results: (0- is unlimited)

Display Type

Dynamic Content Parameter:

Random Output:

Content Type:

of seconds - CacheInterval:

Hide: ☐

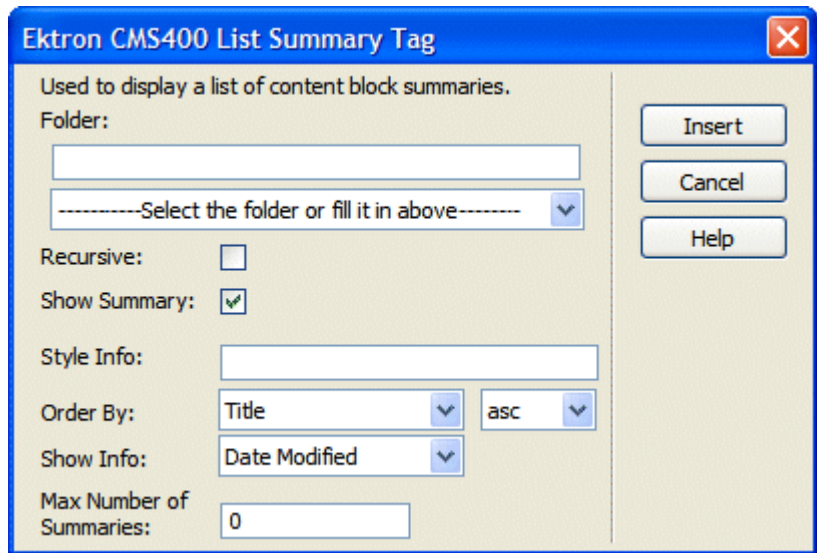
Visible: ☒

(Positioning info - GridLayout) Style: leave blank if none

Wrap tag:

IMPORTANT!

When inserting a list summary using ASP, CF, JSP or PHP, a different dialog box appears.

The image shows a Windows-style dialog box titled "Ektron CMS400 List Summary Tag". It has a blue title bar with a close button (X) in the top right corner. The main area is light beige and contains the following controls: a text box for "Folder:" followed by a dropdown menu with the text "-----Select the folder or fill it in above-----"; a "Recursive:" checkbox which is unchecked; a "Show Summary:" checkbox which is checked; a "Style Info:" text box; "Order By:" with a dropdown menu set to "Title" and a direction dropdown set to "asc"; "Show Info:" with a dropdown menu set to "Date Modified"; and "Max Number of Summaries:" with a text box containing "0". On the right side of the dialog, there are three buttons: "Insert", "Cancel", and "Help".

4. Use the following table to assist you with inserting a metadata control.

Field	Description
Control ID <hr/> <i>Used in ASP.NET only.</i> <hr/>	A unique name for the function you are inserting.
Folder ID	Enter the folder ID. If you do not know the folder path, you can navigate to it using the Select folder drop down box.
Select Folder	Select the folder the summary information comes from. The folder drop-down contains folders from your site. If you insert the folder path, you do not need to select a folder from the dropdown box.
Recursive	If checked, summary information is gathered from the folder selected and all of its child folders.

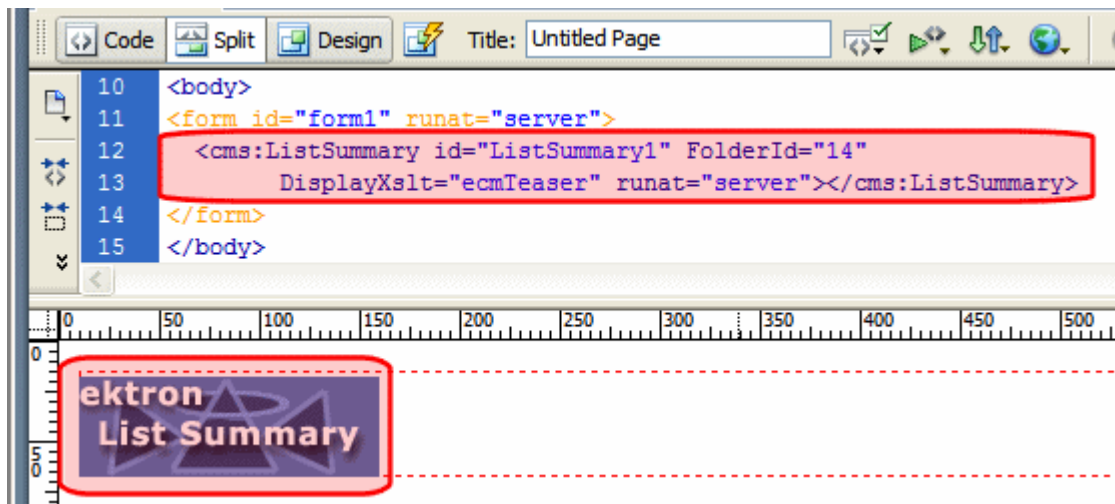
Field	Description
Order By	<p>Specify how the summaries are ordered. Sort by the following criteria.</p> <ul style="list-style-type: none"> • Title - The title of the content block • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • Last Author Last Name - The last author's last name • Last Author First Name - The last author's first name • Ascending - Alphabetic or numerical order • Descending - Reverse alphabetic or numerical order
Max Number of results	Enter a number to specify how many results are displayed on the page (0=unlimited).
Display Type <hr/> Used in ASP.NET only.	<p>Determines how the information on the page is displayed.</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder <i>See Also: "Example of ecmNavigation Display" on page 85</i></p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary <i>See Also: "Example of ecmTeaser Display" on page 85</i></p>
LinkTarget	<p>Defines the way a link acts when a link is clicked.</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>
Dynamic Content Parameter <hr/> Used in ASP.NET only.	<p>Checks the QueryString for this value and replaces the list summary with a content block when specified. Choose None to always display List Summary.</p>

Field	Description
Random Output <hr/> Used in ASP.NET only. <hr/>	Displays random content within a list summary. None - Displays all content in list summary. Random Teaser - Displays a random content block's teaser. Random Body - Displays a random content block's body.
Content Type <hr/> Used in ASP.NET only. <hr/>	Defines what type of list summary appears. Content - Displays only content blocks in the folder. All Types - Displays all content in the folder. Forms - Displays only forms in the folder. Archive_Content - Displays archived content from the folder. Archive_Forms - Displays archived forms from the folder.
# of seconds - CacheInterval <hr/> Used in ASP.NET only. <hr/>	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Hide <hr/> Used in ASP.NET only. <hr/>	Used to hide a list summary in design time and run time. <input checked="" type="checkbox"/> Checked = Hide list summary <input type="checkbox"/> Unchecked = Show list summary <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Visible <hr/> Used in ASP.NET only. <hr/>	Used to show or hide the function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only. <hr/>	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"

Field	Description
Wrap Tag <hr/> Used in ASP.NET only.	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code.

5. When completed, click **insert**.

A graphic defines where the list summary appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<pre> <cms:ListSummary id="ListSummary1" FolderId="14" DisplayXslt="ecmTeaser" runat="server"></cms:ListSummary> </pre>
ASP	<pre> <% ecmListSummary "\Human Resources",1,1,"","DateModified,desc", "DateCreated",0,"","" %> </pre>
ColdFusion	<pre> <cfmodule template="#request.ecm.AppPath#ecmListSummary.cfm" Folder="\Human Resources" MaxNumber="0" Recursive="1" ShowSummary="1" StyleInfo="" OrderBy="DateModified,desc" ShowInfo="DateCreated"> </pre>
JSP	<pre> <%= ecmListSummary("\Human Resources",0,1,"", "Title,asc","DateModified",0,"","") %> </pre>

Platform	Code View
PHP	<pre><?php ecmListSummary("\Human Resources",1,1,"", "DateModified,desc","DateCreated",0,"",""); ?></pre>

Login

For a description of the Login button, see ["Login Server Control" on page 143](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
- **Commands > Ektron CMS400.NET (Server Controls) > CMS Content Controls...**

To insert a login button:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **Login**.
3. The Insert Ektron CMS400.NET Login box appears.

Insert Ektron CMS400 Login Tag

Paints the login or logout button which allows the user to login and use the CMS400 application.

ID:

OnlyAllowMemberLogin: ☐

AutoLogin: ☐

SuppressHelpButton: ☐

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style:

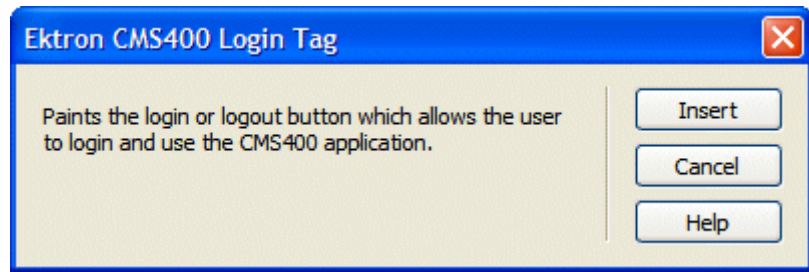
leave blank if none

Wrap tag: ▼

Insert Cancel Help




IMPORTANT!

When inserting a login function using ASP, CF, JSP or PHP, the dialog box that appears has no settings.



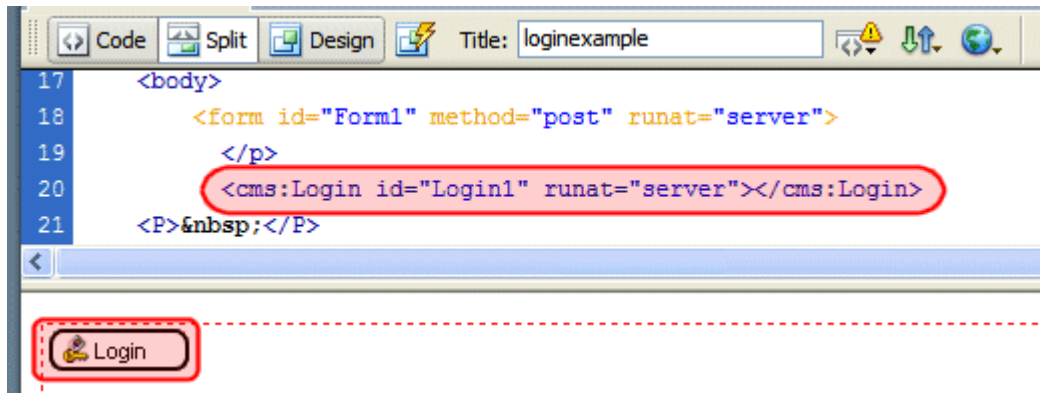
4. Use the following table to assist you with inserting a Login button.

Field	Description
ID <hr/> Used in ASP.NET only.	A unique name for the function you are inserting.
OnlyAllowMemberLogin <hr/> Used in ASP.NET only.	This changes the Login button so only membership users can log in. This prevents someone from trying to login as an Ektron CMS400.NET user and gain access to the Ektron CMS400.NET Workarea. The default is False. <input checked="" type="checkbox"/> Checked = Allow only membership users to login. <input type="checkbox"/> Unchecked = Allow Ektron CMS400.NET users and membership users to login.
AutoLogin <hr/> Used in ASP.NET only.	When set to true, users are automatically logged in using Active Directory authentication. They do not need to enter a username or password. The default is False. <input checked="" type="checkbox"/> Checked = Use Active Directory authentication when logging in. <input type="checkbox"/> Unchecked = Do not use Active Directory authentication when logging in. <hr/> Important!: For this property to function properly, you must be using Active Directory authentication with your CMS400.NET Site.

Field	Description
SuppressHelpButton <hr/> Used in ASP.NET only. <hr/>	Hides the Help button that appears below the Login button when set to true. The default is False. <input checked="" type="checkbox"/> Checked = Do not display the Help button.  <input type="checkbox"/> Unchecked = Display Help button.  
Hide <hr/> Used in ASP.NET only. <hr/>	Used to hide a Login button at run time. <input checked="" type="checkbox"/> Checked = Hide Login button <input type="checkbox"/> Unchecked = Show Login button <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Visible <hr/> Used in ASP.NET only. <hr/>	Used to show or hide the function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden.
Style (Positioning Info - GridLayout) <hr/> Used in ASP.NET only. <hr/>	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"
Wrap Tag <hr/> Used in ASP.NET only. <hr/>	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code.

5. Click **insert**.

A login icon indicates where the Login button appears on the template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:Login id="Login1" runat="server"></cms:Login></code>
ASP	<code><% ecmlogin %></code>
ColdFusion	<code><cfmodule template="#request.ecm.AppPath#ecmLogin.cfm"></code>
JSP	<code><%= ecmLogin() %></code>
PHP	<code><?php ecmlogin() ?></code>

Membership Control

For a description of the Membership function, see ["Membership Server Control" on page 146](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Community Controls...**

To insert a Membership Control:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Membership**.

3. The Membership dialog box appears.

Ektron CMS400 membership

Membership Control.

ID:

DisplayMode:

RedirectFailedURL:

RedirectSuccessURL:

RegisterButtonImg:

RegisterButtonText:

ResetButtonImg:

ResetButtonText:

ShowExtended: ☒

UserExistsMessage:

UserSuccessMessage:

UserUpdateSuccessMessage:

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style:

Wrap tag:

Insert

Cancel

Help

4. Use the following table to assist you with inserting a Membership Control.

Field	Description
ID	A unique name for the function you are inserting.

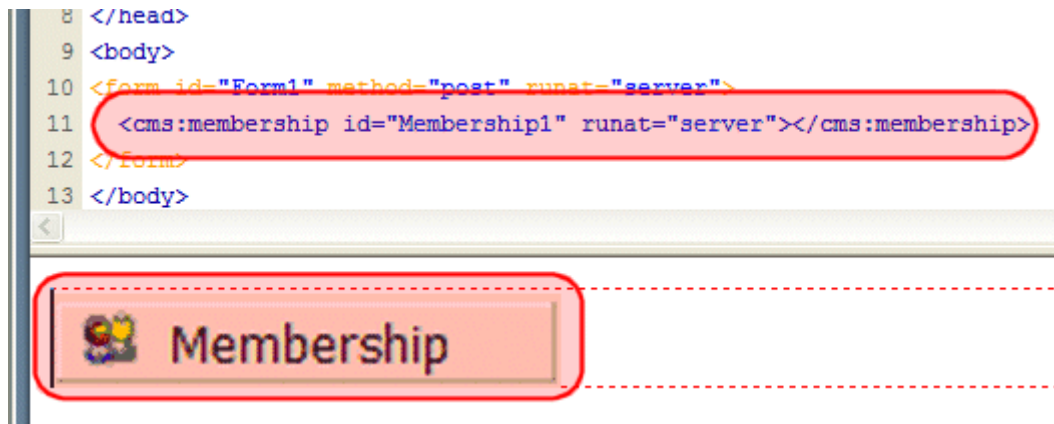
Field	Description
DisplayMode	<p>The type of membership form displayed. Choose from the following:</p> <ul style="list-style-type: none"> • UserRegistration - Lets a site visitor to register as a membership user. • ResetPassword - Lets a membership user to reset his password. • UnsubscribeSecured - Lets a membership user unsubscribe. The membership user needs to enter his username and password to unsubscribe. • UnsubscribeUnsecured - Lets a membership user unsubscribe by entering his membership username only. • AccountActivate - Let a site visitor activate his membership account by entering the ID number sent him in the account verification email. <p>For more information on the DisplayMode property, see "DisplayMode" on page 150.</p>
RedirectedFailedURL	<p>The URL where a membership user is sent if the registration fails. If the page resides in the same folder as the registration page, you only need to enter the name of the page. For example, RegFailed.aspx. If the redirect page is located in a subfolder, you need to add the folder path. For example, members\RegFailed.aspx.</p>
RedirectedSuccessURL	<p>The URL where a membership user is sent if the registration fails. If the page resides in the same folder as the registration page, you only need to enter the name of the page. For example, RegFailed.aspx. If the redirect page is located in a subfolder, you need to add the folder path. For example, members\RegFailed.aspx.</p>
RegisterButtonImg	<p>Lets you add an image in place of the text on the register button.</p>
RegisterButtonText	<p>Text that appears on the Register button. The default is Register. If you use a register button image, you do not see this text.</p>
ResetButtonImg	<p>Lets you add an image in place of the text on the reset button.</p>
ResetButtonText	<p>Text that appears on the reset button. The default is Reset. If you use a reset button image, you do not see this text.</p>

Field	Description
ShowExtended	<p>Decide if the Custom User Properties are available when using this control. The default setting is True.</p> <p>True = Show Custom User Properties</p> <p>False = Hide Custom User Properties</p> <hr/> <p>Note: For more information on Custom User Properties, see the Administration Manual section "Custom User Properties".</p> <hr/>
UserExistsMessage	The message that appears when a membership user already exists. The default message is: Username(email) already exists!
UserSuccessMessage	The message that appears when a membership user successfully registers. The default message is: You have registered successfully.
UserUpdateSuccessMessage	The message that appears when a membership user successfully updates their information. The default message is: You have successfully updated your information.
Hide	<p>Used to hide a Language API in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide</p> <p><input type="checkbox"/> Unchecked = Show Language API</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **insert**.

A graphic defines where the Membership Control appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:membership id="Membership1" runat="server"></cms:membership></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Metadata for ASP.NET Pages

For a description of the Metadata function, see ["MetaData Server Control" on page 167](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a metadata function:

1. Move the blinking cursor to the area that you want to place it.

NOTE The Ektron CMS400.NET Metadata function should be inserted within the <head> tags of your template to work properly.

2. From the list of Ektron CMS400.NET Functions, click on **Metadata**.
3. The Insert Ektron Metadata box is displayed.

4. Use the following table to assist you with inserting metadata.

Field	Description
Control ID	A unique name for the function you are inserting.
Content ID	Enter a valid content ID in the Content ID textbox. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content Block drop down boxes.
GenerateDublinCore	<p>When enabled, this property automatically creates seven of the Simple Dublin Core metadata fields from standard CMS400.NET system properties. The default is false.</p> <p><input checked="" type="checkbox"/> Checked = Generate Simple Dublin Core metadata fields</p> <p><input type="checkbox"/> Unchecked = Do not generate Simple Dublin Core metadata fields</p> <p>The seven fields and how they are associated with the CMS400.NET properties is explained in "Using the Simple Dublin Core Metadata Standard" on page 168</p>

Field	Description
Select Category	The drop-down list contains a list of content folders. If you insert the content ID, you do not need to select a category.
Select Content Block	The drop-down box is populated with a list of all the content block titles within that folder. When a title is selected, the content ID textbox is filled with that content block's id number. If you insert the content ID, you do not need to select a content block.
Dynamic Content Parameter	Gets or sets the QueryString parameter to read a content ID dynamically.

5. When completed, click **insert**.

6. The following line appears in the <body> of the HTML code.

```
<cms:MetaData id="MetaData2" runat="server" DefaultContentID="12"></cms:MetaData>
```

7. Move that line from the <body> tag into the <head> tag.

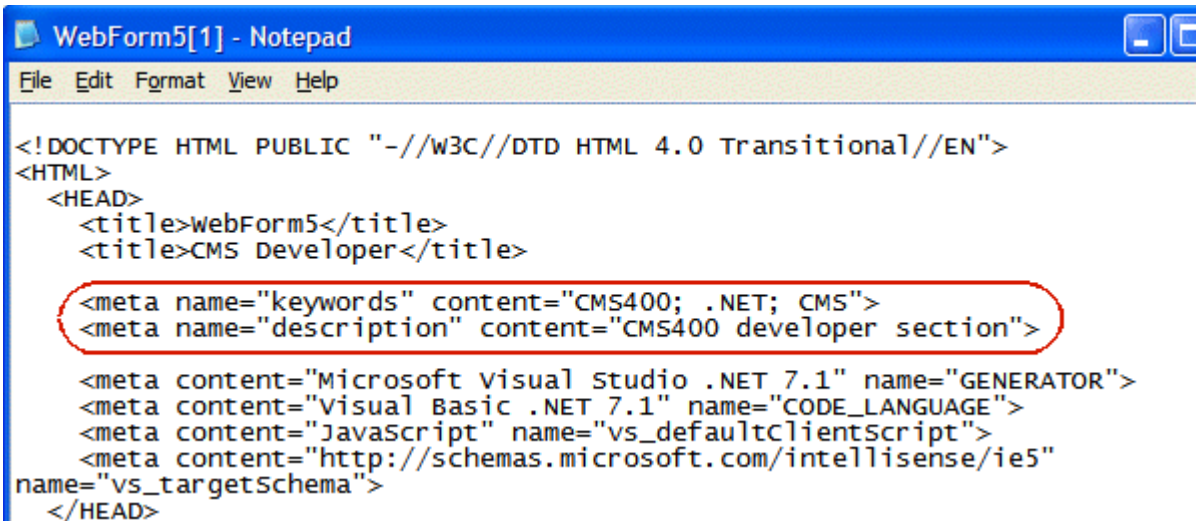
```
<HEAD>
  <title>WebForm5</title>
  <cms:MetaData id="Metadadata2" runat="server" DefaultContentID="12"></cms:MetaData>
  <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
  <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
  <meta name="vs_defaultClientScript" content="JavaScript">
  <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/i
</HEAD>
<body>
  <form id="Form1" method="post" runat="server">
  </form>
</body>
```

8. Save the Web form.

9. Open the Web page in the browser.

10. Right click on the Web page and click **View Source**. Look in the head tag. The meta tags from the content block are added to the page.

The metadata information added is shown below.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>webForm5</title>
    <title>CMS Developer</title>
    <meta name="keywords" content="CMS400; .NET; CMS">
    <meta name="description" content="CMS400 developer section">
    <meta content="Microsoft Visual Studio .NET 7.1" name="GENERATOR">
    <meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
```

Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:MetaData id="MetaData2" runat="server" DefaultContentID="12"></cms:MetaData></code>
ASP	For ASP pages, see "Metadata Function for ASP, CF, JSP or PHP Pages" on page 362
ColdFusion	For ColdFusion pages, see "Metadata Function for ASP, CF, JSP or PHP Pages" on page 362
JSP	For JSP pages, see "Metadata Function for ASP, CF, JSP or PHP Pages" on page 362
PHP	For PHP pages, see "Metadata Function for ASP, CF, JSP or PHP Pages" on page 362

Metadata Function for ASP, CF, JSP or PHP Pages

For a description of the Metadata function, see ["MetaData Server Control" on page 167](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a metadata function:

1. Move the blinking cursor to the area that you want to place it.

NOTE The Ektron CMS400.NET Metadata function should be inserted within the <head> tags of your template to work properly.

2. From the list of Ektron CMS400.NET Functions, click on **Metadata**.
3. The Insert Ektron Metadata box is displayed.

4. Use the following table to assist you with inserting metadata.

Field	Description
Metadata List	When you select a content ID and click the Add to Metadata List button, The content ID is added to this text box. This allows you to create a list of content blocks for which the metadata of each content block is added to the page.
Content ID	Enter a valid content ID in the Content ID textbox. If you do not know the Content ID, you can navigate to it using the Select Category and Select Content Block drop down boxes.

Field	Description
GenerateDublinCore	<p>When enabled, this property automatically creates seven of the Simple Dublin Core metadata fields from standard CMS400.NET system properties. The default is false.</p> <p><input checked="" type="checkbox"/> Checked = Generate Simple Dublin Core metadata fields</p> <p><input type="checkbox"/> Unchecked = Do not generate Simple Dublin Core metadata fields</p> <p>The seven fields and how they are associated with the CMS400.NET properties is explained in "Using the Simple Dublin Core Metadata Standard" on page 168</p>
Select Category	The drop-down list contains a list of content folders. If you insert the content ID, you do not need to select a category.
Select Content Block	The drop-down box is populated with a list of all the content block titles within that folder. When a title is selected, the content ID textbox is filled with that content block's id number. If you insert the content ID, you do not need to select a content block.
Metadata types for exclusion	Select the metadata types that you do not want added to the metadata function you are inserting. Hold the control key down for multiple selections.

5. When completed, click **insert**.

6. The following line appears in the <body> of the HTML code.

```
<cms:MetaData id="MetaData2" runat="server" DefaultContentID="12"></cms:MetaData>
```

7. Move that line from the <body> tag into the <head> tag.

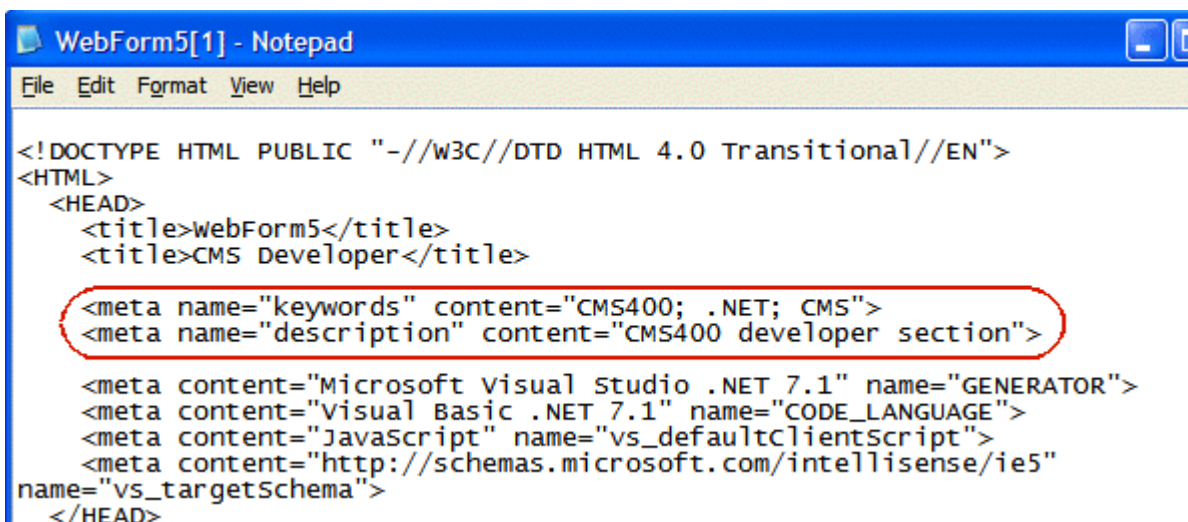
```
<HEAD>
  <title>WebForm5</title>
  <cms:MetaData id="Metadadata2" runat="server" DefaultContentID="12"></cms:MetaData>
  <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
  <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
  <meta name="vs_defaultClientScript" content="JavaScript">
  <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/i
</HEAD>
<body>
  <form id="Form1" method="post" runat="server">
  </form>
</body>
```

8. Save the Web form.

9. Open the Web page in the browser.

10. Right click on the Web page and click **View Source**. Look in the head tag. The meta tags from the content block are added to the page.

The metadata information added is shown below.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>webForm5</title>
    <title>CMS Developer</title>
    <meta name="keywords" content="CMS400; .NET; CMS">
    <meta name="description" content="CMS400 developer section">
    <meta content="Microsoft Visual Studio .NET 7.1" name="GENERATOR">
    <meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
```

Shown below is an example of the code that is inserted:

Platform	Code View
.NET	For ASP.NET pages, see "Metadata for ASP.NET Pages" on page 359
ASP	<% ecmMetadata "[1;Title][9][3;Keywords,Title]", "" %>
ColdFusion	<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="[1;Title][9][3;Keywords,Title]">
JSP	<%= ecmMetadata("[12;keywords,Title]", "") %>
PHP	<?php ecmMetadata("[1;Title][9][3;Keywords,Title]", ""); ?>

Multipurpose Content Block

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a Multipurpose content block function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Multipurpose Tag...**

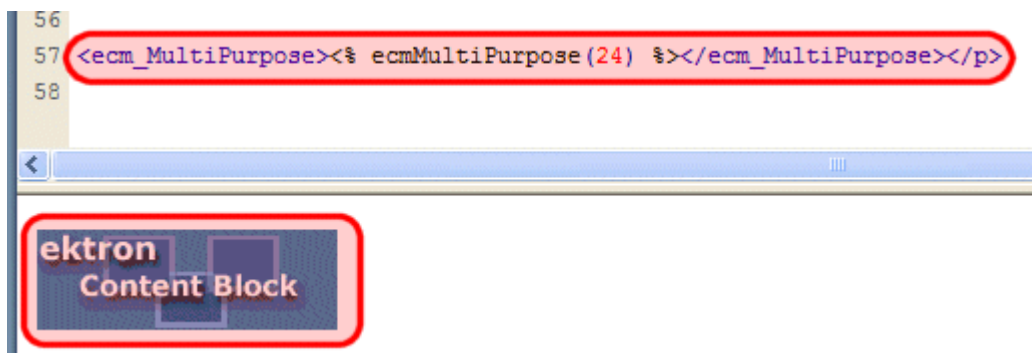
3. The Insert Ektron CMS400.NET Multipurpose Tag box is displayed.

4. Use the following table to assist you with inserting a Multipurpose tag.

Field	Description
Default Content ID	Type in the default content ID, or select a category from the drop down list, then select a content block from the second dropdown list.

5. When completed, click **insert**.

A graphic to define where the Multipurpose tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
.NET	In ASP.NET, the Content Block Tag functions as a Multipurpose Content block.
ASP	<code><% ecmMultiPurpose(24) %></code>
ColdFusion	<code><cfmodule DefaultID="24" template="#request.ecm.AppPath#ecmMultipurpose.cfm"></code>

Platform	Code View
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	<code><?php ecmMultiPurpose(12); ?></code>

Poll

For a description of the Poll function, see ["Poll Server Control" on page 176](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a Poll:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click **Poll**
3. The Insert Poll box is displayed.

4. Use the following table to assist you with inserting a form.

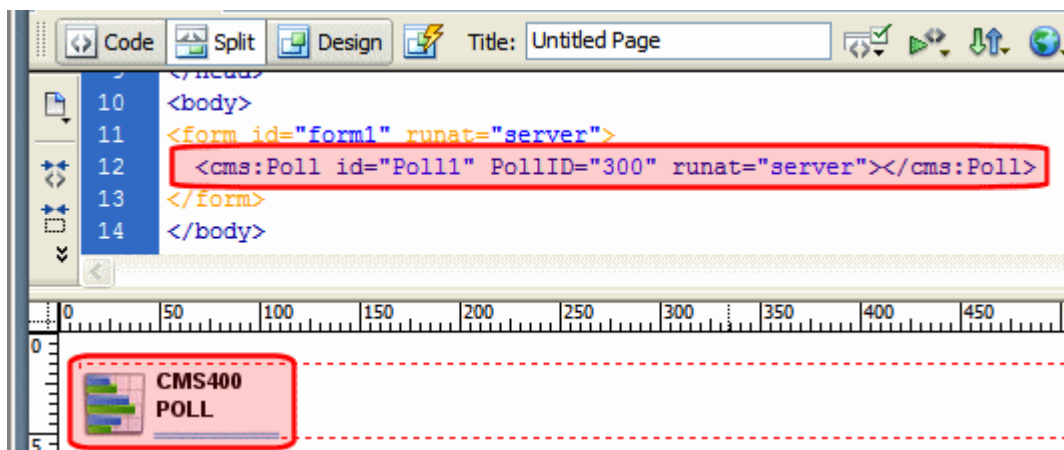
Field	Description
Control ID	A unique name for the function you are inserting.

Field	Description
Form ID	Type in the form ID that you want to insert. If you do not know the Form ID, you can navigate to it using the Select Form drop down box.
Select Form	The drop-down box is populated with a list of all the form titles. When a title is selected, the form ID textbox is filled with that form's id number. If you insert the Form ID, you do not need to select a form.
Dynamic Form Parameter	To make this form block dynamic, select ekfrm . When you do, this control uses the form block passed as a URL parameter.
Ajax	<p>Displays the poll or results, using an iFrame, in the area of the page that contains the poll without disturbing its surrounding contents. The default is true.</p> <p><input checked="" type="checkbox"/> Checked = Polls and results are shown in an iFrame without any modification or refresh of the surrounding contents.</p> <p><input type="checkbox"/> Unchecked = The poll and the results replace the Web page's content.</p>
# of seconds - CacheInterval	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.
Add Validation	When checked , adds Validation to your main runat=server form.
Include Form Tags	Determines if tags are generated automatically or manually. See Also: "Automatic versus Manual Generation of Form Tags" on page 119
Hide.	<p>Used to hide a form in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide form</p> <p><input type="checkbox"/> Unchecked = Show form</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>

Field	Description
Style (Positioning Info - GridLayout)	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"
Wrap Tag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code.

- When completed, click **Insert**.

A graphic defines where the form appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:Poll id="Poll1" PollID="300" runat="server"></cms:Poll></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Random Content

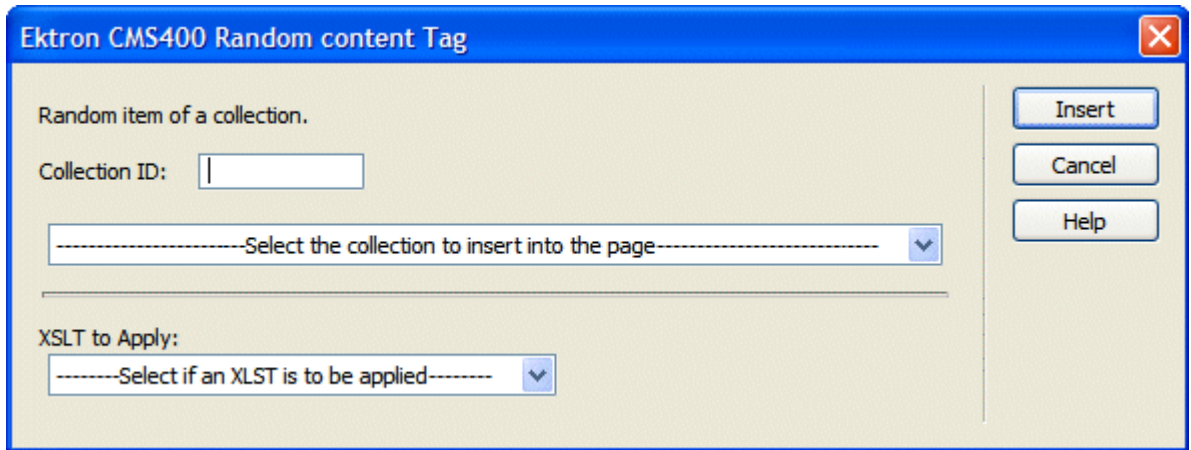
For a description of the Random Content function, see ["Random Content" on page 476](#)

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a Random Content function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Insert Random Content Tag...**
3. The Insert Random Content Tag box is displayed.



4. Use the following table to assist you with inserting a Random Content tag.

Field	Description
Collection ID	Type in the collection ID, or select it from the drop down list, of the collection where the random content function will retrieve content from.
XSLT to apply <i>XML Content only</i>	If desired, you may specify an XSLT to apply to the content that gets displayed.

5. When completed, click **Insert**.

A graphic to define where the Random Content tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
.NET	This function is not available as an Ektron Dreamweaver Extension for .NET
ASP	<% ShowRandomContent 3, 0 %>
ColdFusion	<cfmodule id="3" displayFunction="0" template="#request.ecm.AppPath#ecmShowRandomContent.cfm">
JSP	<%= ShowRandomContent(3, 0) %>
PHP	<?php ShowRandomContent(3, 0); ?>

Random Summary

For a description of the Random Summary function, see ["Random Summary" on page 477](#)

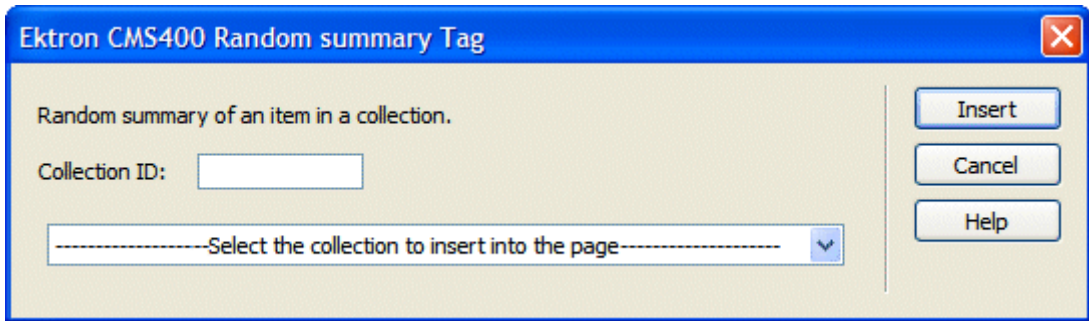
This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a Random Summary function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Random Summary Tag...**

3. The Insert Ektron CMS400.NET Random Summary Tag box is displayed.



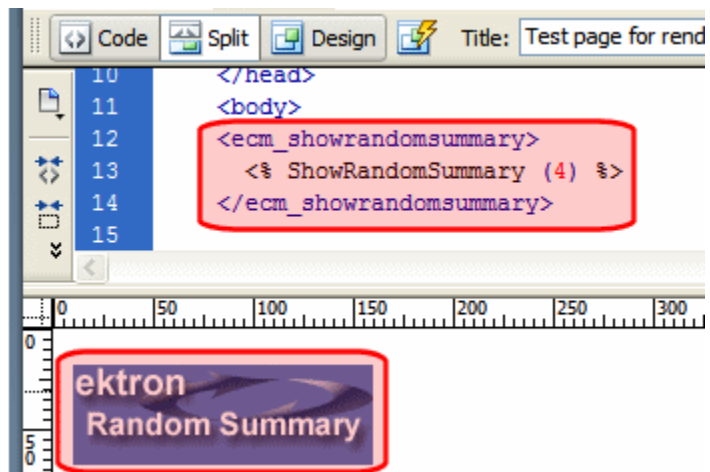
The dialog box is titled "Ektron CMS400 Random summary Tag". It contains the text "Random summary of an item in a collection." Below this is a "Collection ID:" label followed by a text input field. Underneath the input field is a dropdown menu with the placeholder text "-----Select the collection to insert into the page-----". On the right side of the dialog, there are three buttons: "Insert", "Cancel", and "Help".

4. Use the following table to assist you with inserting a Random Summary tag.

Field	Description
Collection ID	Type in the collection ID, or select it from the drop down list, of the collection where the random summary function will retrieve summaries from.

5. When completed, click **Insert**.

A graphic to define where the Random Summary tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
.NET	This function is not available as an Ektron Dreamweaver Extension for .NET
ASP	<code><% ShowRandomSummary (2) %></code>

Platform	Code View
ColdFusion	<code><cfmodule id="2" template="#request.ecm.AppPath#ecmShowRandomSummary.cfm"></code>
JSP	<code><%= ShowRandomSummary(2) %></code>
PHP	<code><?php ShowRandomSummary (2); ?></code>

RSS Aggregator

For a description of the RSS Aggregator function, see ["RssAggregator Server Control" on page 179](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Control...**

To insert an RSS Aggregator:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **RSS Aggregator**.
3. The RSS Aggregator dialog box appears.

Ektron CMS400 RSS Aggregator

Used to display an RSS feed.

Control ID:

URL:

Max Number of results:

Display Type:

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style:

Wrap tag:

leave blank if none

Buttons: Insert, Cancel, Help

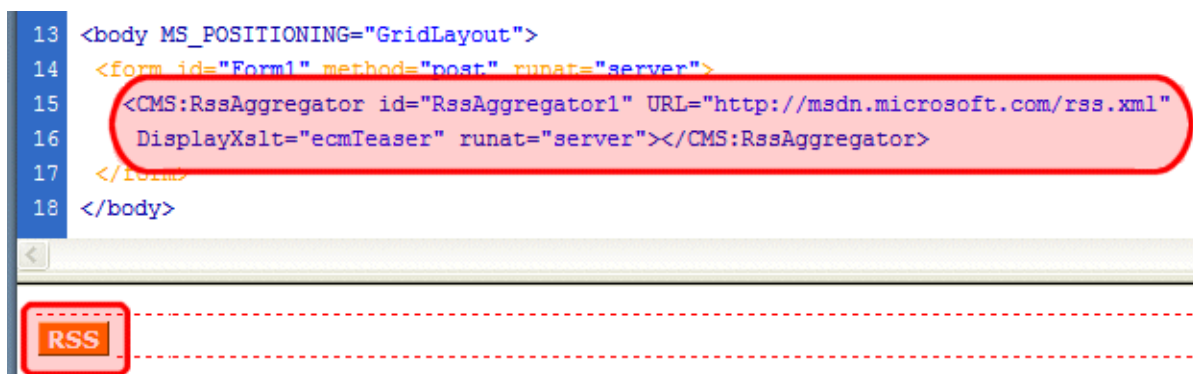
4. Use the following table to assist you with inserting an RSS Aggregator.

Field	Description
Control ID	A unique name for the function you are inserting.
URL	The RSS feed path for the server control. For example: <code>http://msdn.microsoft.com/rss.xml</code>
Max Number of Results	The Maximum number of items from an RSS feed that are returned (0=unlimited).
Display Type	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every RSS feed item See Also: "Example of ecmNavigation Display" on page 85 ecmTeaser - lists a title and a description of every RSS feed item See Also: "Example of ecmTeaser Display" on page 85 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page
Hide	Used to hide a Language API in design time and run time. <input checked="" type="checkbox"/> Checked = Hide <input type="checkbox"/> Unchecked = Show Language API <hr/> Note: If Hide and Visible are both checked, the property is hidden. <hr/>
Visible	Used to show or hide the function at run time. <input checked="" type="checkbox"/> Checked = Hide function <input type="checkbox"/> Unchecked = Show function <hr/> Note: If Hide and Visible are both checked, the property is hidden. <hr/>
Style (Positioning Info - GridLayout)	Set position information of a control as absolute, when using a GridLayout Web page. For example: style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"

Field	Description
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **insert**.

A graphic defines where the RSS Aggregator appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<CMS:RssAggregator id="RssAggregator1" URL="http://msdn.microsoft.com/rss.xml" DisplayXslt="ecmTeaser" runat="server"></CMS:RssAggregator>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Search

For a description of the Search functions, see ["Search Server Control" on page 182](#)

This tag appears on the following menu(s):

-
- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**
or
 - **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

To insert a Search function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Search**

3. The Insert Ektron Search box is displayed.

Insert Ektron CMS400 Search

Search ----- Control ID: Recursive: ☒ AJAX ☒

FolderID:

-----Select the or fill it in above-----

Order By: Ascending

Display Type:

Max Number of results: (0- is unlimited)

ShowSearchBoxAlways: ☒ Display Horizontal: ☐ LinkTarget

RemoveTeaserHtml ☒ Hide: ☐ Visible: ☒ Wrap tag:

ResultTagId PostResultsAt

ButtonImgSrc: ButtonText:

(Positioning info - GridLayout) Style: leave blank if none

EmptyResultMsg: MaxCharacters: TextBoxSize: (If 0 ignore)

EnableAdvancedLink ☒ EnableArchived ☐ EnableArchivedBtn ☐

EnableLibHyperlinks ☒ EnableLibHyperlinksBtn ☒ EnableLibFiles ☒

EnableLibFilesBtn ☒ EnableLibImages ☒ EnableLibImagesBtn ☒

EnableDMSAssets ☒ EnableDMSAssetsBtn ☒ EnableForms ☒

EnableFormsBtn ☒ EnableContent ☒ EnableContentBtn ☒

EnableContentFieldset ☐ EnableFieldset ☐ ShowSearchOptions ☐

TargetList ShowExtendedSearch ☒

IMPORTANT!

When inserting a list summary using ASP, CF, JSP or PHP, a slightly different dialog box appears.

4. Use the following table to assist you with inserting a search.

Field	Description
Ajax	<p>Set to true to enable Ajax searches. When enabled, the <code>MaxResults</code> property determines the maximum number of results per page, and the <code>ResultTagId</code> property becomes active.</p> <p><input checked="" type="checkbox"/> Checked = Enable Ajax Search</p> <p><input type="checkbox"/> Unchecked = Original HTML Search</p> <p>See Also: "Ajax Search" on page 197</p>
ButtonImgSrc (ASP.NET) Button Image Source (ASP, CF, JSP or PHP)	<p>If you wish to use an image for the search button, insert the location of the image in this field. If no image is specified, a standard submit button is used.</p> <p>See Also: "Button Image Source Examples" on page 199</p>
ButtonText	<p>The button text is the text that is displayed on the standard submit button, or the alt text of a search button image.</p> <p>See Also: "Search Button Text Examples" on page 200</p>

Field	Description
Control ID <hr/> Used in ASP.NET only. <hr/>	A unique name for the function you are inserting.
Display Horizontally	The search can be displayed either vertically or horizontally on your Web page. Check off this box if you want it to appear horizontally. See Also: "Display Examples" on page 201
Display Type <hr/> Used in ASP.NET only. <hr/>	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every content block in the folder ecmTeaser - lists the title of every content block in the folder plus the content summary See Also: "Example of ecmNavigation Display" on page 85 and "Example of ecmTeaser Display" on page 85
EmptyResultMsg <hr/> Used in ASP.NET only. <hr/>	The return message displayed, if the search returns no hits. The default is: Search returned zero results
EnableAdvancedLink <hr/> Used in ASP.NET only. <hr/>	Set to true to display the link that makes the advanced properties available. <input checked="" type="checkbox"/> Checked = Display Advanced link <input type="checkbox"/> Unchecked = Hide Advanced link See Also: "Using the Advanced Search Properties" on page 206
EnableArchived <hr/> Used in ASP.NET only. <hr/>	Enables users to search archived content. The archived content must be archived as Archived Remain On Site. The default is False. <input checked="" type="checkbox"/> Checked = Searching archived content is enabled. <input type="checkbox"/> Unchecked = Searching archived content is disabled, unless Include Archived is checked in the Advanced search. See Also: "Using the Advanced Search Properties" on page 206

Field	Description
EnableArchivedBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform an archive search. The default is False.</p> <p><input checked="" type="checkbox"/> Checked = The Include Archive check box is enabled in the advanced search properties. If used with EnableArchived, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Include Archive check box is disabled in the advanced search properties.</p>
EnableContent <hr/> Used in ASP.NET only.	<p>Enables users to search content. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching content is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching content is disabled, unless Content is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>
EnableContentBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a content search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The Content check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Content check box is disabled in the advanced search properties.</p>
EnableContentFieldset <hr/> Used in ASP.NET only.	<p>Creates a border around the group of content items in an advanced search. These content items include Content, Forms and DMS Assets.</p> <p>The default is False.</p> <p><input checked="" type="checkbox"/> Checked = Show border around content items.</p> <p><input type="checkbox"/> Unchecked = No border around content items.</p> <p>See Also: "Fieldset Properties" on page 212</p>
EnableDMSAssets <hr/> Used in ASP.NET only.	<p>Enables users to search DMS Assets. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching DMS Assets is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching DMS Assets is disabled, unless DMS Assets is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>

Field	Description
EnableDMSAssetsBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a DMS Asset search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The DMS Asset check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The DMS Asset check box is disabled in the advanced search properties.</p>
EnableForms <hr/> Used in ASP.NET only.	<p>Enables users to search Forms. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching Forms is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching Forms is disabled, unless Forms is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>
EnableFormsBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a Forms search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The Forms check box is enabled in the advanced search properties. If used with EnableContent, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Forms check box is disabled in the advanced search properties.</p>
EnableLibFiles <hr/> Used in ASP.NET only.	<p>Enables users to search Library Files. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching Library Files is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching Library Files is disabled, unless Files is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>
EnableLibFilesBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a Library Files search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The Files check box is enabled in the advanced search properties. If used with EnableLibFiles, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Files check box is disabled in the advanced search properties.</p>

Field	Description
EnableLibHyperlinks <hr/> Used in ASP.NET only.	<p>Enables users to search Library Hyperlinks. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching Library Hyperlinks is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching Library Hyperlinks is disabled, unless Hyperlinks is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>
EnableLibHyperlinksBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a Library Hyperlinks search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The Hyperlinks check box is enabled in the advanced search properties. If used with EnableLibHyperlinks, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Hyperlinks check box is disabled in the advanced search properties.</p>
EnableLibImages <hr/> Used in ASP.NET only.	<p>Enables users to search Library Images. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = Searching Library Images is enabled.</p> <p><input type="checkbox"/> Unchecked = Searching Library Images is disabled, unless Images is checked in the Advanced search.</p> <p>See Also: "Using the Advanced Search Properties" on page 206</p>
EnableLibImageBtn <hr/> Used in ASP.NET only.	<p>Provides a check box so users can decide if they want to perform a Library Images search. The default is Unchecked.</p> <p><input checked="" type="checkbox"/> Checked = The Images check box is enabled in the advanced search properties. If used with EnableLibImages, the check box is checked by default.</p> <p><input type="checkbox"/> Unchecked = The Images check box is disabled in the advanced search properties.</p>
Folder	Type in the folder path to be searched. If you do not know the folder path, you can navigate to it using the Select folder drop down box.
Font Color <hr/> Used in ASP, CF, JSP and PHP only.	The color of the font used in the search function.

Field	Description
Font Face <hr/> Used in ASP, CF, JSP and PHP only. <hr/>	The type of font to use for search function.
Font Size <hr/> Used in ASP, CF, JSP and PHP only. <hr/>	The size of the font used in the search function.
Hide <hr/> Used in ASP.NET only. <hr/>	Used to hide a search in design time and run time. <input checked="" type="checkbox"/> Checked = Hide search <input type="checkbox"/> Unchecked = Show search <hr/> Note: If Hide and Visible are both checked, the property is hidden. <hr/>
LinkTarget <hr/> Used in ASP.NET only. <hr/>	Defines the way a link acts when a link is clicked. The choices are: _blank - This target causes the link to always be loaded in a new blank window. This window is not named. _self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. _parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent. _top - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.
MaxCharacters (ASP.NET) Text Box Max Char (ASP, CF, JSP or PHP)	Specify the maximum amount of characters that is accepted in the text box.
Max Number of results <hr/> Used in ASP.NET only. <hr/>	Enter a number to specify how many results are displayed on the page (0=unlimited). When the Ajax property is set to checked, the way this property works changes from the maximum amount of results to the maximum results per page. This takes advantage of Ajax's ability to have paging in the searches. If zero is entered, all results appear on the same page.

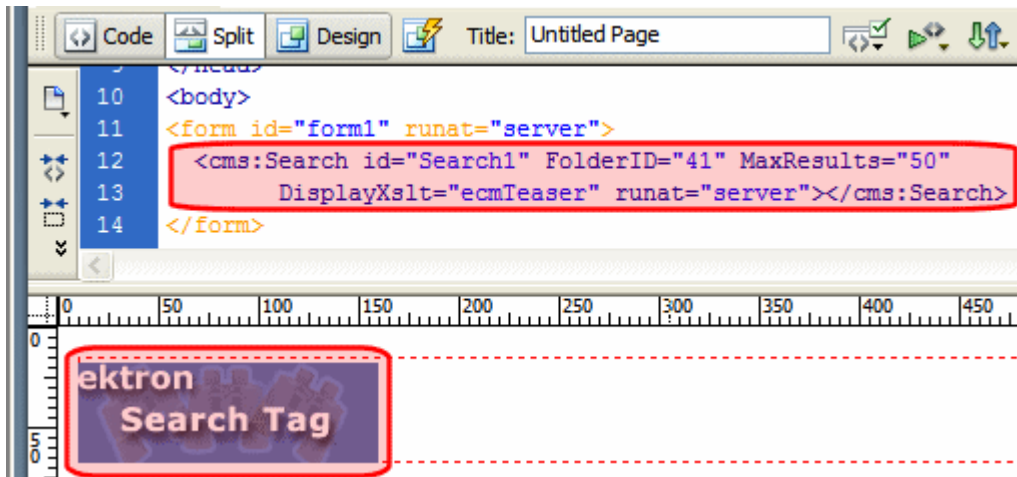
Field	Description
Order By <hr/> Used in ASP.NET only. <hr/>	Specify how you want to order the search results that are generated when a search is performed on you Web site. Also specify if you want the results in ascending or descending order. <ul style="list-style-type: none"> • Title - The title of the content block • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • Last Author Last Name - The last author's last name • Last Author First Name - The last author's first name • Ascending - Alphabetic or numerical order • Descending - Reverse alphabetic or numerical order
PostResultsAt (ASP.NET) Target Page (ASP, CF, JSP or PHP)	The path of the Web page where the search results will be posted. The # symbol appears in the text box by default. The # symbol tells the control to post back to the Web page that created the search. <hr/> Important: A search control must exist on the posting page. <hr/>
Recursive	If checked, search information is gathered from the folder selected and all of its child folders.
RemoveTeaserHtml <hr/> Used in ASP.NET only. <hr/>	An optional search result formatting parameter that removes all the HTML from each content item's teaser/summary before sending it to the browser. <p><input checked="" type="checkbox"/> Checked = Remove HTML</p> <p><input type="checkbox"/> Unchecked = Do not remove HTML</p>
ResultTagId <hr/> Used in ASP.NET only. <hr/>	Specify the div tag control ID to display the search results. This property is only active when the EnableAjax property is set to true. See Also: " ResultTagId " on page 198
Select Folder	Select a folder path to be searched. If you insert the folder path, you do not need to select a folder.

Field	Description
ShowExtendedSearch	<p>Allows a user to search via the metadata type: Searchable Property.</p> <p><input checked="" type="checkbox"/> Checked = Show extended search.</p> <p><input type="checkbox"/> Unchecked = Hide extended search.</p> <p>See Also: "ShowExtendedSearch" on page 202</p> <p>For more information on Metadata and Searchable Property, see the Administrator Manual section "Metadata".</p>
ShowSearchBoxAlways <hr/> <i>Used in ASP.NET only.</i> <hr/>	<p>If unchecked, the search box does not appear onPostBack.</p>
ShowSearchOptions <hr/> <i>Used in ASP.NET only.</i> <hr/>	<p>If set to true, the following choices appear along with the search box. This type of search is also known as a Standard search. The default setting is Checked.</p> <ul style="list-style-type: none"> • All the words • Any of the words • Exact phrase • Match Partial words <p><input checked="" type="checkbox"/> Checked = Show search options</p> <p><input type="checkbox"/> Unchecked = Hide search options</p> <p>See Also: "Basic and Standard Searches" on page 208</p>
Style (Positioning Info - GridLayout) <hr/> <i>Used in ASP.NET only.</i> <hr/>	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>

Field	Description
TargetFolderDescriptionList	<p>A comma delineated list of folders to search. The target folder list uses the Description property of a folder to define which folders are searched. This is an optional property. If you use this property, it will supersede the FolderID property. The default is blank. Below is an example of the syntax used.</p> <p>Football=1,News\Headlines=0</p> <p>Football, News and Headlines are the folder descriptions. The ones (1) and zeros (0) define whether a site visitor is forced to search the folder or not.</p> <p>1 = Automatically search this folder 0 = Do not automatically search this folder</p> <p>See Also: "TargetFolderDescriptionList" on page 205</p>
TextBoxSize	<p>Specify the size of the text box that is displayed on the Web page for users to enter the search keywords. The width is based on the number of characters.</p> <p>See Also: "Text Box Examples" on page 199</p>
Visible <hr/> Used in ASP.NET only.	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p>
Wrap Tag <hr/> Used in ASP.NET only.	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>

- When completed, click **insert**.

A graphic defines where the search appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:Search id="Search1" FolderID="41" MaxResults="50" DisplayXslt="ecmTeaser" runat="server"></cms:Search></code>
ASP	<code><% ecmSearch "\Products",1,"SearchDisplay.asp",10,25,"","Search", "Verdana", "Blue", "2",0,"" %></code>
ColdFusion	<code><CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm" StartingFolder="\Products" Recursive="1" TargetPage="SearchDisplay.cfm" TextBoxSize="10" MaxCharacters="25" ButtonImageSrc="" ButtonText="Search" FontFace="Verdana" FontColor="Blue" FontSize="2" Horizontal="0"></code>
JSP	<code><%= ecmSearch("\Products",1,"SearchDisplay.jsp",10,25,"","Search", "Verdana", "Blue", "2",0,"") %></code>
PHP	<code><?php ecmSearch("\Products",1,"SearchDisplay.php",10,25,"","Search", "Verdana", "Blue", "",0,""); ?></code>

Search Display

The search display custom tag creates the area where search results appear after a search is executed.

This tag appears on the following menu(s):

- Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a search display function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Insert Search Display Tag...**
3. The Insert Ektron Search Display Tag box is displayed.

Ektron CMS400 Search Display Tag

Creates area where search results will be displayed.

Maximum Number of Documents: (0=Unlimited)

Style Info:

Show Last Modified Date: ☐

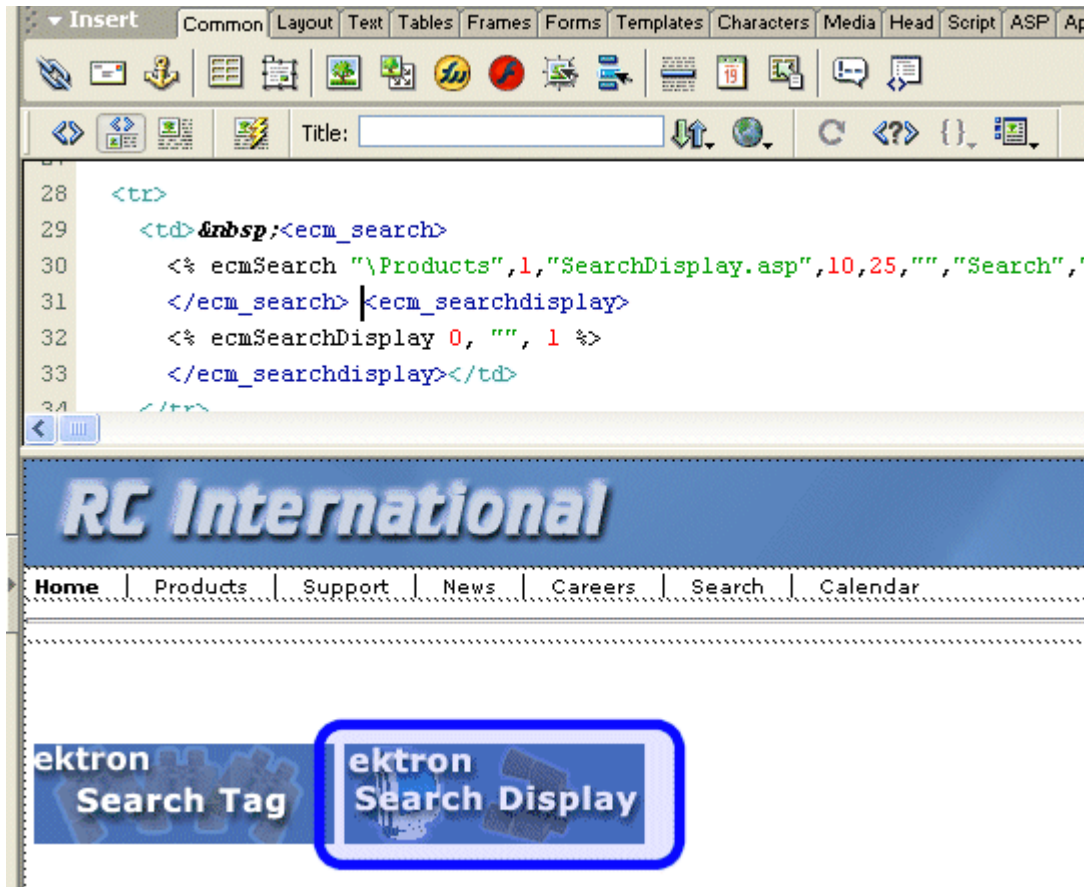
Insert
Cancel
Help

4. Use the following table to assist you with inserting a search display function.

Field	Description
Maximum Number of Documents	Specify the maximum amount of results that will be returned to the user when a search is performed.
Style Info	Insert an inline style sheet that will be used to format all the hyperlinks that are returned.
Show Last Modified Date	Check off this box to show the last modified date with the search result. This allows users to distinguish between updated and old content.

- When completed, click **insert**.

A graphic to define where the search display tag has been placed on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View (with no default content block set)
.NET	In ASP.NET, you do not need to add a separate search display.
ASP	<% ecmSearchDisplay 0, "", 1 %>
ColdFusion	<cfmodule template="#request.ecm.AppPath#ecmSearchDisplay.cfm" MaxNumber="0" StyleInfo="" ShowDate="1">
JSP	<%= ecmSearchDisplay(0, "", 1)%>
PHP	<?php ecmSearchDisplay(0, "", 1)?>

Single Summary

For a description of the Single Summary function, see ["Single Summary Function" on page 456](#)

This tag appears on the following menu(s):

- **Commands > Ektron CMS400 - (ASP CF JSP PHP) > CMS Function...**

To insert a Single summary:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Content functions, click on **Single Summary Tag...**
3. The Insert Ektron Single Summary Tag box is displayed.

Ektron CMS400 Single Summary Tag

Used to display a content block summary.

Content ID:

-----Select the category to retrieve list of content-----

-----Select the content block to insert into the page-----

Show Summary: ☒

Style Info:

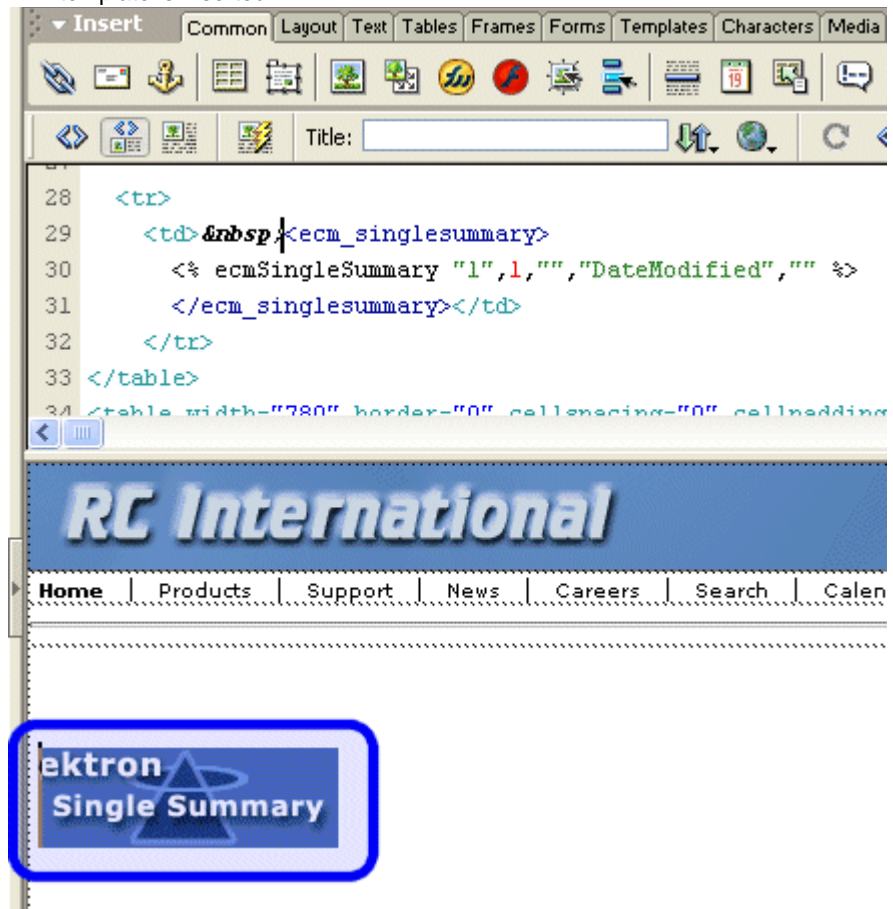
Show Info:

4. Use the following table to assist you with inserting a single summary tag.

Field	Description
Content ID	<ul style="list-style-type: none">• Working Offline - Enter a valid content ID into the Content ID textbox.• Working Online - The top drop-down list will be filled with a list of content folders that you have privileges to. After choosing a content folder, the bottom drop-down box will be filled with a list of all the content block titles within that category. When a title is selected, the content ID textbox will be filled with that content block's id number.
Show Summary	If checked, content summaries will be displayed.
Style Info	Insert any inline style information, which will be applied to the generated text hyperlink, here.
Show Info	Select the type of information that will be displayed next to the generated hyperlinks.

- When completed, click **insert**.

A graphic to define where the single summary tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ecmSingleSummary "1",1,"","DateModified","" %>
ColdFusion	<cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm" ContentID="1" ShowSummary="1" StyleInfo="" ShowInfo="DateModified">
JSP	<%= ecmSingleSummary("1",1,"","DateModified","") %>
PHP	<?php ecmSingleSummary("1",1,"","DateModified","") ?>

Site Map

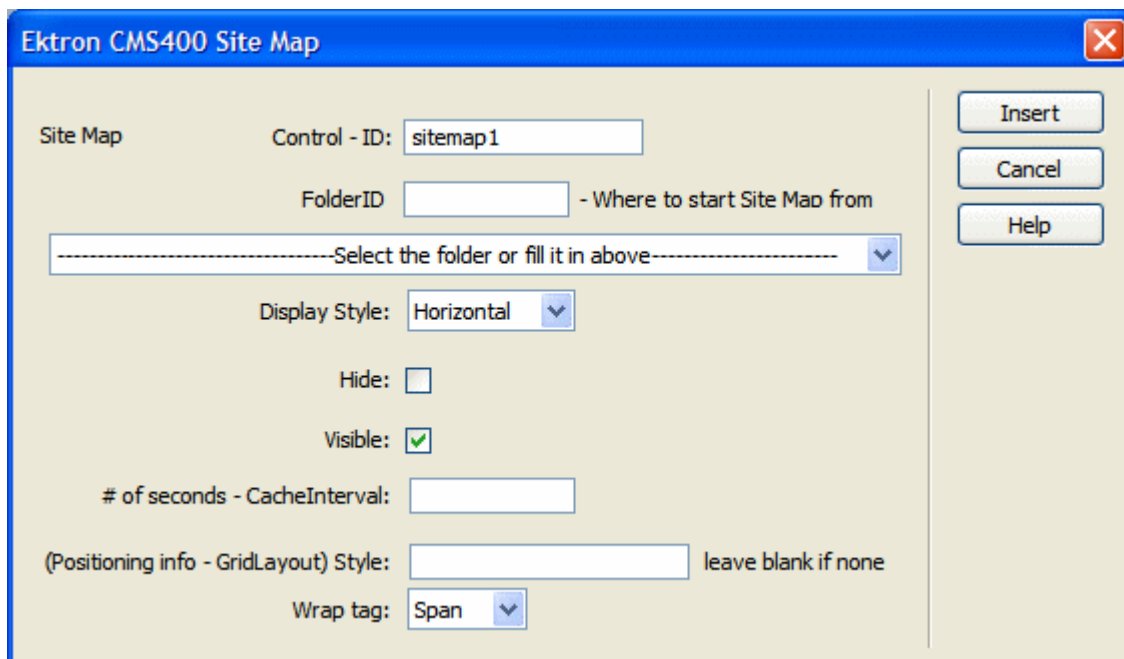
For a description of the Site Map function, see ["SiteMap Server Control" on page 235](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Controls...**

To insert a Site Map:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Site Map**.
3. The Site Map dialog box appears.



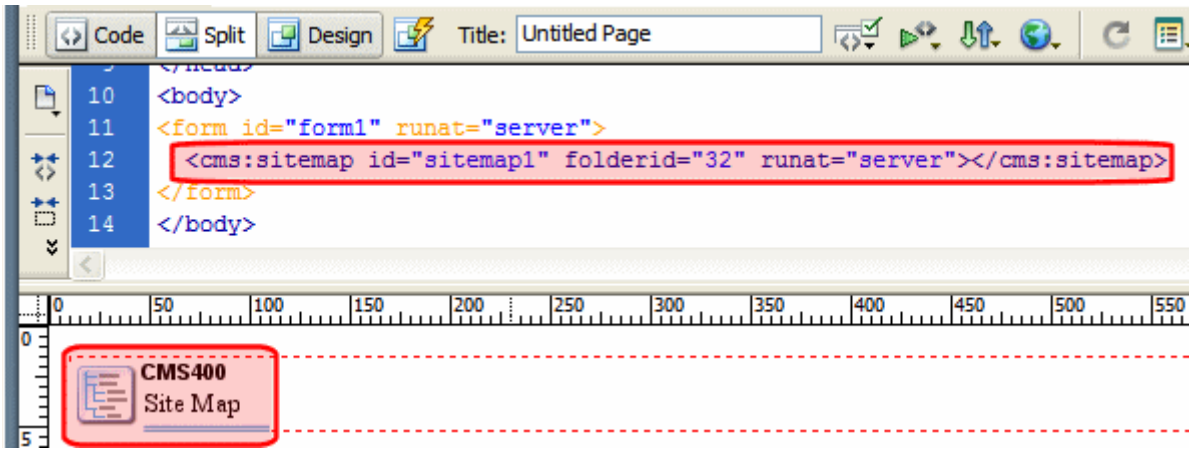
4. Use the following table to assist you with inserting a Folder Bread Crumb.

Field	Description
Control ID	A unique name for the function you are inserting.
FolderID	The numeric id of the folder that you want to use. If you do not know the folder ID, you can navigate to it using the Select folder drop down box.
DisplayStyle	Indicate how to display the folder bread crumb: horizontally or vertically. The default is Horizontal.

Field	Description
Hide	<p>Used to hide a Language API in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide</p> <p><input type="checkbox"/> Unchecked = Show Language API</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
# of seconds - CacheInterval	<p>Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36.</p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.</p>

5. When completed, click **Insert**.

A graphic defines where the Site Map appears on your template.



Shown below is an example of the code that is inserted:

Platform	Code View
.NET	<code><cms:sitemap id="sitemap1" folderid="32" runat="server"></cms:sitemap></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Standard Menu

For a description of the Standard Menu function, see ["Menu Server Control" on page 152](#).

This tag appears on the following menu(s):

- **Commands > Ektron CMS400.NET (Server Controls) > CMS Module Controls...**

To insert a Standard Menu:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET Functions, click **Standard Menu**

The Standard Menu Control box appears.

Ektron CMS400 Menu Control

Menu

Control ID:

DefaultMenuID:

-----Select the from a list of menus-----

MenuType:

Dynamic menu Parameter:

of seconds - CacheInterval:

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style: leave blank if none

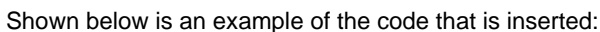
Wrap tag:

3. Use the following table to assist you with inserting a Standard Menu.

Field	Description
Control ID	A unique name for the function you are inserting.
DefaultMenuID	The numeric id of the menu that you want to insert. If you do not know the DefaultMenuID, you can navigate to it using the Select Menu drop down box.
Select Menu	Select a menu to be used. When a menu is selected, the DefaultMenuID textbox is filled with that menu's id number. If you insert the DefaultMenuID, you do not need to select a menu.
MenuType	The type of menu to render. None - databind only SampleMenu - A sample display, formatted as a bulleted menu list TreeMenu - A sample display, formatted as a folder tree. You can expand the tree by clicking on the folder icon. For more information on using the SampleMenu and TreeMenu Display, see "Using DisplayXslt Samples" on page 154.

Field	Description
Dynamic Menu Parameter	<p>Checks the QueryString for the menu value and replaces the menu with content when ID is specified. Leave blank to always display the menu.</p> <p>None - Use Default</p> <p>menu_ID - The control uses the menu passed as a URL parameter.</p>
CacheInterval	<p>Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the <code>CacheInterval</code> property to 300 (three hundred). See Also: "Caching with Server Controls" on page 36</p>
Hide	<p>Used to hide a menu in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p>Note: If Hide and Visible are both checked, the property is hidden.</p> <hr/>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page.</p> <p>For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The <code></code> tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <code><div></code> tag is used when you want to apply attributes to a block of code.</p>

- A graphic defines where the Standard Menu appears on your template.



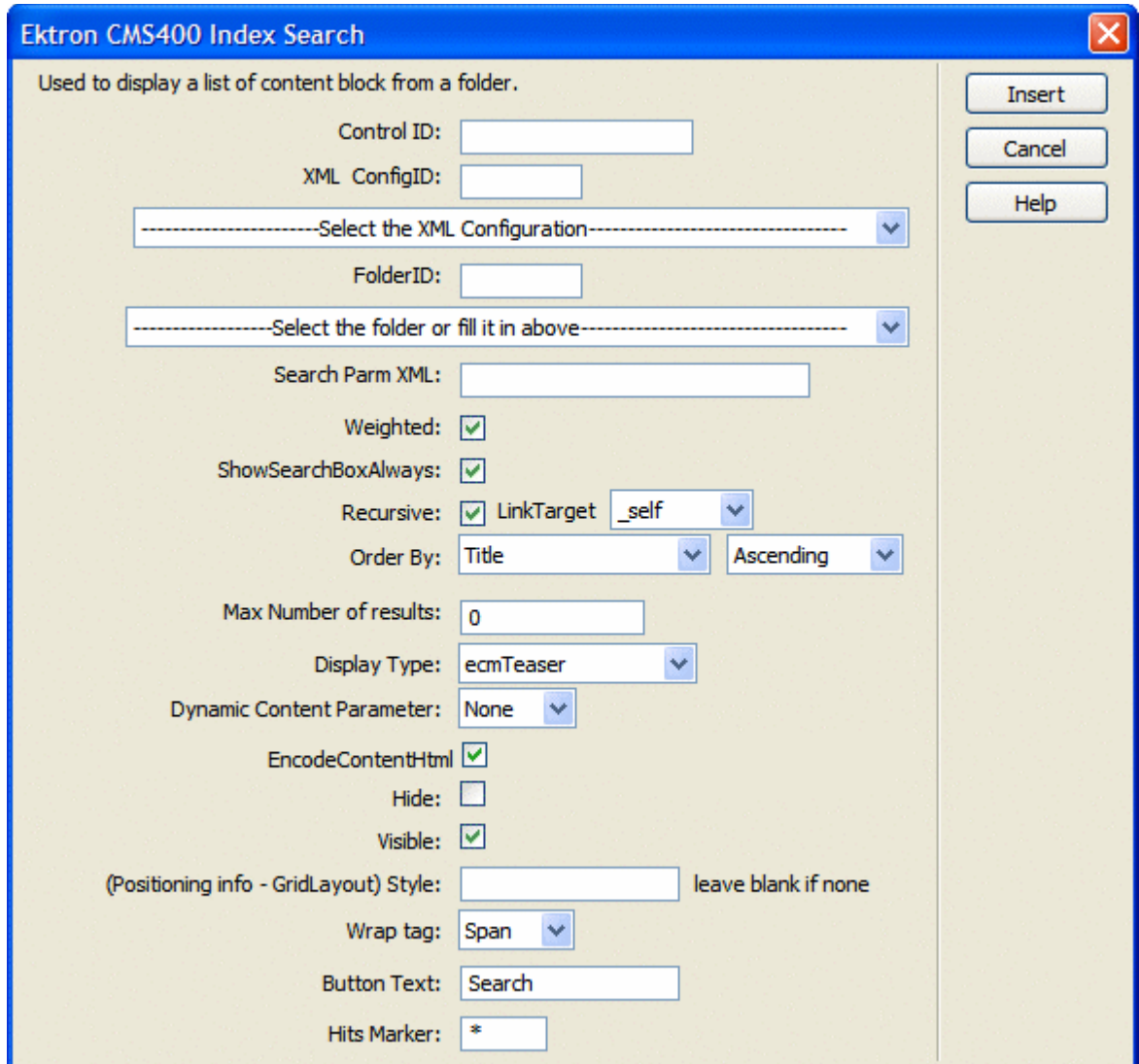
Platform	Code View
.NET	<cms:Menu id="Menu1" DefaultMenuID="15" runat="server"></cms:Menu>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

For a description of the XML Index Search Function, see "IndexSearch Server Control" on page 218.

- **Commands > Ektron CMS400.NET (Server Controls) > Content Controls...**

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS400.NET functions, click **XML Index Search**.

3. The Insert XML Index Search box is displayed.



Ektron CMS400 Index Search

Used to display a list of content block from a folder.

Control ID:

XML ConfigID:

-----Select the XML Configuration-----

FolderID:

-----Select the folder or fill it in above-----

Search Parm XML:

Weighted: ☒

ShowSearchBoxAlways: ☒

Recursive: ☒ LinkTarget

Order By:

Max Number of results:

Display Type:

Dynamic Content Parameter:

EncodeContentHtml ☒

Hide: ☐

Visible: ☒

(Positioning info - GridLayout) Style: leave blank if none

Wrap tag:

Button Text:

Hits Marker:

4. Use the following table to assist you with inserting an XML Index Search.

Field	Description
Control ID	A unique name for the function you are inserting.
XML ConfigID	ID of the Smart Form whose indexed fields appear on the search screen. If you do not know the XML ConfigID, you can navigate to it using the Select Smart Form drop down box.

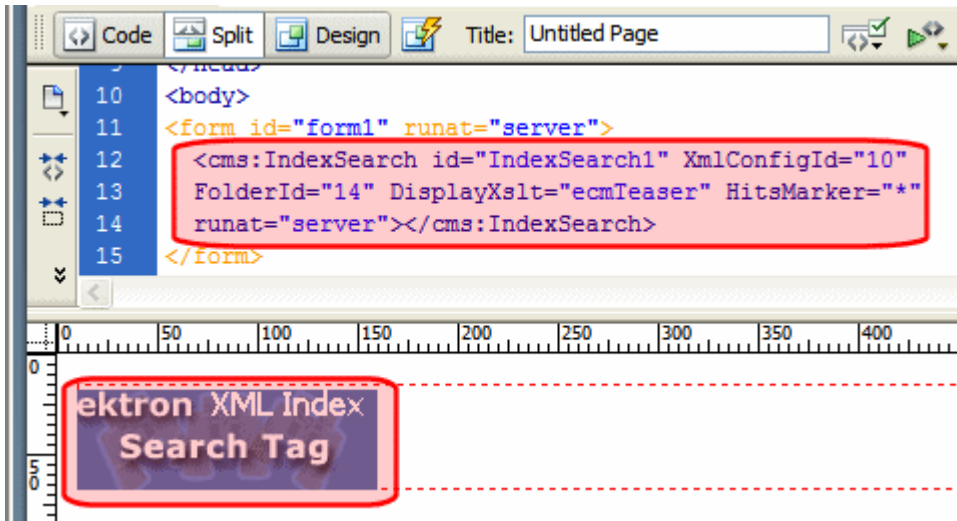
Field	Description
Select XML Configuration	The drop-down box is populated with a list of all the XML Configuration titles. When a title is selected, the XML ConfigID textbox is filled with that XML Configuration's id number. If you insert the XML ConfigID, you do not need to select a XML Configuration.
FolderID	The numeric id of the folder that you want to search. If you do not know the folder path, you can navigate to it using the Select folder drop down box.
Select Folder	Select a folder path to be searched. When a folder is selected, the folderID textbox is filled with that folder's id number. If you insert the Folder ID, you do not need to select a folder.
Search Parm XML	This property preserves settings that a developer would create in Visual Studio.
Weighted	<p>If this value is checked, content blocks are ranked by the number of criteria matches.</p> <p>For example, if a content block matches all search criteria, it appears at the top of the results page. A hit marker appears for each match.</p> <p>If a second content block only matches one criterion, it also appears on the results page underneath the first content block. One asterisk appears above the page's title to indicate the single match.</p> <p><input checked="" type="checkbox"/> Checked- display every content block that matches <i>at least one</i> search criteria</p> <p><input type="checkbox"/> Unchecked - show only content blocks that match <i>all</i> search criteria</p>
ShowSearchBoxAlways	If set to unchecked , the search box does not appear on when the results are posted to the Web page.
Recursive	Determines whether the XML Index Search Control uses this folder's child folders in the search.
LinkTarget	<p>Defines the way a link acts when a link is clicked.</p> <p>_blank - This target causes the link to always be loaded in a new blank window. This window is not named.</p> <p>_self - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.</p> <p>_parent - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.</p> <p>_top - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.</p>

Field	Description
Order By	<p>Specify how you want to order the search results that are generated when a search is performed on you Web site. Also specify if you want the results in ascending or descending order.</p> <ul style="list-style-type: none"> • Title - The title of the content block • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • Last Author Last Name - The last author's last name • Last Author First Name - The last author's first name • Ascending - Alphabetic or numerical order • Descending - Reverse alphabetic or numerical order <p>When you choose weighted search results, Ascending and Descending use how many criteria match your search. For example, if you have weighted results, and you choose Ascending, the results that match the most criteria appear first.</p>
Max Number of results	The Maximum number of content blocks returned (0=unlimited).
Display Type	<p>Determines how the information on the page is displayed</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder</p> <p>See Also: "Example of ecmNavigation Display" on page 85</p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary</p> <p>See Also: "Example of ecmTeaser Display" on page 85</p>
Dynamic Content Parameter	<p>Checks the QueryString for the content block ID value and replaces the search with content when ID is specified. Leave blank to always display the search. By making content dynamic, when the content displayed on a Web page contains a link to another piece of content, you can display the second piece of content in the same Web page space.</p> <p>None - Use Default</p> <p>ID - The control uses the content block passed as a URL parameter.</p>
EncodeContentHtml	<p>When checked, Content HTML is encoded when it is returned.</p> <p><input checked="" type="checkbox"/> Checked = Content HTML is encoded</p> <p><input type="checkbox"/> Unchecked = Content HTML is not encoded</p>

Field	Description
Hide	<p>Used to hide a form in design time and run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide form</p> <p><input type="checkbox"/> Unchecked = Show form</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Visible	<p>Used to show or hide the function at run time.</p> <p><input checked="" type="checkbox"/> Checked = Hide function</p> <p><input type="checkbox"/> Unchecked = Show function</p> <hr/> <p><u>Note: If Hide and Visible are both checked, the property is hidden.</u></p>
Style (Positioning Info - GridLayout)	<p>Set position information of a control as absolute, when using a GridLayout Web page. For example:</p> <p>style="Z-INDEX: 101; LEFT: 328px; POSITION: absolute; TOP: 240px"</p>
Wrap Tag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p>
Button Text	<p>The text that appears on the button.</p>
Hits Marker	<p>This symbol shows how many search criteria each hit matched.</p> <p>The following example shows the first hit met three of the criteria. The second group met one criterion.</p> <div data-bbox="696 1248 1056 1537" data-label="Image"> </div>

- When completed, click **insert**.

A graphic defines where the XML Index Search appears on your template.



Shown below is an example of the code that is inserted.

Platform	Code View
.NET	<code><cms:IndexSearch id="IndexSearch1" XmlConfigId="10" FolderId="14" DisplayXslt="ecmTeaser" HitsMarker="*" runat="server"></cms:IndexSearch></code>
ASP	This function is not available as an Ektron Dreamweaver Extension for ASP.
ColdFusion	This function is not available as an Ektron Dreamweaver Extension for ColdFusion.
JSP	This function is not available as an Ektron Dreamweaver Extension for JSP.
PHP	This function is not available as an Ektron Dreamweaver Extension for PHP.

Standard ASP.NET Methods

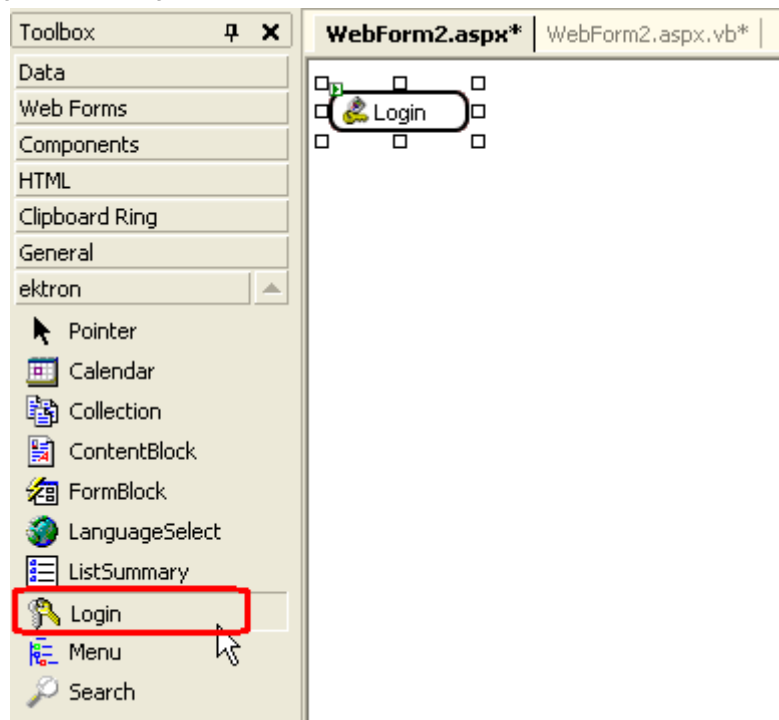
Use these methods to customize **Ektron CMS400.NET**. To use them, insert them onto your template pages to invoke content blocks, list summaries, and other objects stored in your database. They are the primary building blocks of your **Ektron CMS400.NET** Content Management System.

Replacement of Standard Methods

Beginning with Release 4.7, **Ektron CMS400.NET** utilizes ASP.NET server controls, which replace most Standard ASP.NET methods. With **Ektron CMS400.NET**'s new toolbox of pre-built, reusable, data-bindable ASP.NET controls, Ektron empowers developers working in Visual Studio 2005 to rapidly integrate Ektron CMS components into their templates and sites. For more Information on Ektron CMS400.NET controls, see ["Introduction to Ektron CMS400.NET Server Controls" on page 7](#).

The following examples shows the Login server control added by drag and drop vs. adding a Login method by the ASP.NET method.

Drag and Drop Login server control



Adding a Login function via an ASP.NET method.

```
<tr>
  <td>
    <%=AppUI.ecmLogin()%>
```

```
</td>  
</tr>
```

Best Practice

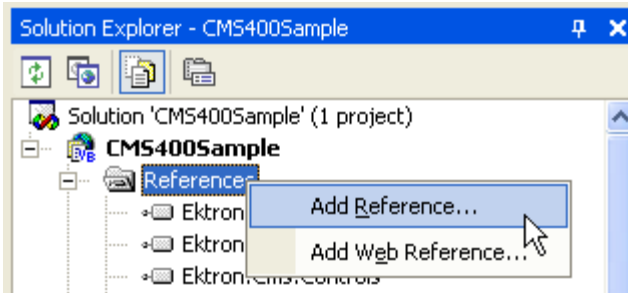
Ektron recommends using CMS400.NET server controls when available over Ektron standard ASP.NET methods. The methods are being replaced by the server controls.

You can still use the standard ASP.NET methods in **Ektron CMS400.NET**.

Referencing Methods in Visual Basic

Before you can use methods with Visual Basic, you must add a reference to them from Visual Basic. Follow these steps to do so.

1. Open Visual Basic.
2. Choose or add a project.
3. In Solution Explorer, right click **References** and choose **Add Reference**.

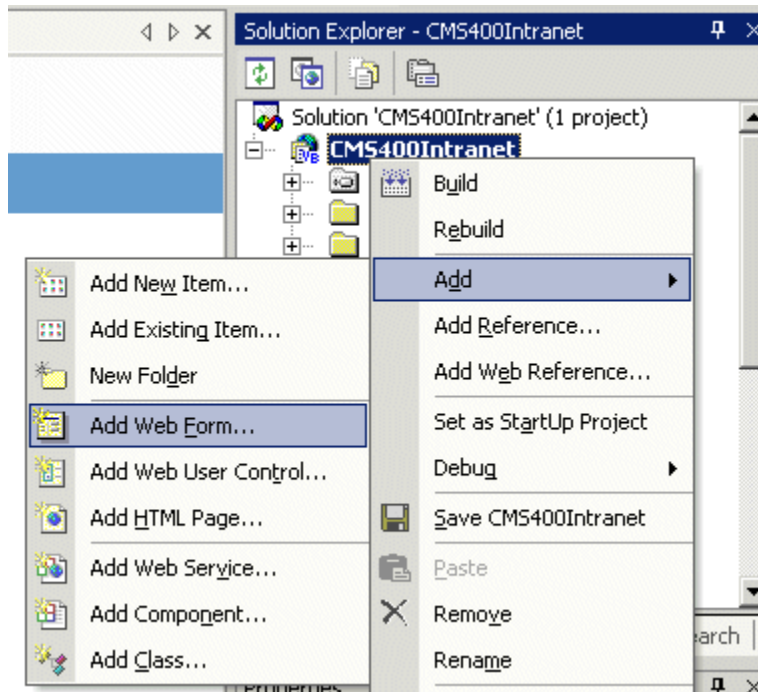


4. In the Add Reference dialog box, click **Browse**.
5. Go to `Inetpub/wwwroot/cms400demo/Bin`.
6. To load the standard methods click `Ektron.Cms.UI.CommonUI.dll`. To load the XML Indexing methods, click `Ektron.Cms.Common.dll`.
7. Click **Open**, then **OK**.

Adding a New Web Form

To add a new Web form, follow these steps.

1. Open your project in Visual Studio 2005.
2. Right click the project.
3. Click **Add > Add Web Form**.



4. Click **Web Form**.
5. In the code behind, add an Imports `Ektron.Cms.UI.CommonUI` statement.

Requirements for Standard Methods

Ektron CMS400.NET maintains and displays content through the use of Ektron ASPX methods and ASP.NET assembly objects. To use the methods, include the following code in your template:

```
<%@ Page Language="vb" AutoEventWireup="false" %>
<%@ Import Namespace="Ektron.Cms.UI.CommonUI" %>
<% Dim AppUI As New ApplicationAPI %>
```

Each line is explained below.

Line	Description
<%@ Page Language="vb" AutoEventWireup="false" %>	A generic .net line of code that should always be placed in the template
<%@ Import Namespace="Ektron.Cms.UI.CommonUI" %>	An import statement that calls the namespace <code>Ektron.CMS.UI.CommonUI</code> to display information from the .net assembly file

Line	Description
<% Dim AppUI As New ApplicationAPI %>	Creates an instance of the ApplicationUI class that is used by each method

These lines call the Ektron .net assembly object to allow the page to function properly. They must appear at the top of each template file.

Standard Methods

To customize your Ektron CMS400.NET application, you need to be familiar with these methods.

Method	Description	More Information
Add Content	Programmatically adds a content block(s) to Ektron CMS400.NET	"Addcontent" on page 407
Collection	Displays links generated by collections.	"Collection Server Control" on page 82
Content Block	Displays and manages a content block.	"ContentBlock Server Control" on page 91
Event Calendar	Displays an event calendar	"Calendar Server Control" on page 75
Form	Displays a content block that is associated with a form.	"FormBlock Server Control" on page 117
IndexSearch	Displays on a Web page a set of search fields. A site visitor uses the page to search for information on your Web site.	"IndexSearch Server Control" on page 218
List Summary	Displays a list of content block summaries.	"ListSummary Server Control" on page 135
Login	Displays the login or logout button, which lets the user log in and use Ektron CMS400.NET.	"Login Server Control" on page 143
Metadata	Displays the metadata in the source of the Web page displaying the content.	"MetaData Server Control" on page 167
Menu	Customizes the DHTML menus feature	"DhtmlMenu Server Control" on page 160

Method	Description	More Information
Search	Inserts a text box with a search button in the template. When the user completes the search screen, it looks through XHTML content on the Web site.	"Search Server Control" on page 182
ShowSiteLanguage	Lets a site visitor select a language in which to view site.	"LanguageSelect Server Control" on page 133

Addcontent

The addcontent method lets you programmatically add a new content block to Ektron CMS400.NET without logging in. This method can be used for several applications, such as

- A Web form allowing you to collect data from users about the content block
- A loop function that moves information from a database to an Ektron CMS400.NET database

IMPORTANT! Every string that you pass to the function must be HTML encoded. Here is the VB.Net syntax for encoding a string.

```
myTitle = "John's Book"
myTitle = Server.HtmlEncode(myTitle)
```

Explained in the table below are the attributes for the add content block function.

Attribute	Description
Username	The username of the user to be authenticated for permissions to see whether the user is able to add a content block.
Password	The password for the username being authenticated.
Domain	If using Active Directory, specify the domain needed to authenticate the user.
Content_title	Specify a title for the content block being added.
Content_comment	Specify a historical comment for the content being added.
ContentHTML	Actual content for the content block.
SummaryHTML	Summary information about the content block.
ContentLanguage	Language of content.

Attribute	Description
FolderID	ID number of the content folder where the content block will be added to.
GoLive	<p>Date and time the content will go live.</p> <hr/> <p>Must match date/time format set in the Web.config key ek_dateFormat.</p> <hr/>
EndDate	<p>Date and time the content will expire.</p> <hr/> <p>Must match date/time format set in the Web.config key ek_dateFormat.</p> <hr/>
MetaInfo	<p>Specify the information for the metadata which will be used as a reference point for finding other words or information.</p> <p>An example is:</p> <pre><metadata><meta id="3">Title</meta></metadata></pre> <p>Where ID is the metadata definition ID whose value is being set. The value is set between the meta tags and the meta tag is repeatable for as many metadata definitions as you want to set for the given content block.</p>
ErrString	If there is a error then errString contains the error, otherwise errString is empty.

Add Content Example

The following is an example of the add content function in a template:.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Dim m_apiObj As Ektron.Cms.UI.CommonUI.ApplicationAPI
    Dim newContentId As Integer = 0
    Try
        m_apiObj = New Ektron.Cms.UI.CommonUI.ApplicationAPI
        newContentId = m_apiObj.AddContent( _
            "admin", _
            "admin", _
            "", _
            "Content Title", _
            "Comment - this content added through API!", _
            "This is the <b>body</b> of the content.", _
            "This is the <i>summary</i> for the content.", _
            "1033", _
            "0", _
            "06-27-2006 4:09 PM", _
            "", _
            "<metadata><meta id=""3"">ContentTitle</meta></metadata>", _
            "" )
        Literal1.Text = "Content Added, ID = " + newContentId.ToString
    Catch ex As Exception
        Literal1.Text = "Error, AddContent Failed: " + ex.Message
    Finally
        m_apiObj = Nothing
    End Try
End Sub
```

The previous example has the following properties:

- The user “admin” is used to be authenticated.
- The title of the new content block is “Content Title”.
- The comment for the content is “Comment”.
- The body of the content is “This is the **body** of the content”.
- The summary of the content block is “This is the *summary* for the content.”
- The language ID = 1033, or English.
- The folder ID = 0 is where the content is added.
- The start date is 06-27-2006.

- No end date is specified.
- The metadata definition ID being set is "3" and the value is ContentTitle.

When this function is executed, the content is added to Ektron CMS400.NET. Displayed here is the record of the content in the workarea.

View Content "Content Title"

View: English (U.S.) | Add: -

Properties | Content | Summary | Metadata | Comment | Tasks | We

Properties

Content Title:	Content Title
Content ID:	490
Content Language	English (U.S.)
Status:	Approved (Published)
Last User To Edit:	Application Administrator
Last Edit Date:	6/28/2006 2:35:44 PM
Start Date:	6/27/2006 4:09:00 PM
End Date:	[None Specified]
Action on End Date	[None Specified]
Date Created:	6/28/2006 2:35:40 PM
Approval Method:	Do Not Force All Approvers
Approvals:	[None Specified]
Smart Form Configuration:	[None Specified] HTML Content Assumed
Template	dynamic.aspx
Path	\
Rating	this content has not been rated.
Content Searchable:	True

Custom ASP Functions

Ektron CMS400.NET displays content on your Web site through the use of custom Ektron ASP functions and ComObjects. Your template you must include the following file to be able to use the custom functions:

```
<!-- #include file="applicationAPI.asp" -->
```

This file (**application.asp**) calls the file with the custom Ektron functions to allow the page to function properly. The `include` statement must appear at the top of every template file that uses an Ektron CMS400.NET custom ASP function.

IMPORTANT!

`ContentWS.asmx` is the web services (WSDL) file that the custom ASP functions use to talk to **Ektron CMS400.NET**. The WSDLpath located in the `applicationAPI.asp` file must be set to location of the WSDL file. The default WSDLpath is `http://localhost/cms400demo/WorkArea/WebServices/ContentWS.asmx`. If you have changed the location of `ContentWS.asmx` you need to update the WSDLpath. For example, if your site is `www.myelektronsite.com`, the WSDLpath would be `http://www.myelektronsite.com/WorkArea/WebServices/ContentWS.asmx`.

To customize Ektron CMS400.NET, you need to be familiar with these tags.

Function	Description	More Information
Calendar Functions		
Calendar	Displays a calendar with different display types. For example, a daily view calendar.	"Calendar Function" on page 481
Event Calendar	Displays a month view event calendar	"Event Calendar Function" on page 478
Collection Functions		
Collection	Displays links generated by collections	"Collection Function" on page 465
CollectionXML	Creates XML for a collection by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML Collection Function" on page 471
Random Content	Randomly displays the content of a collection item	"Random Content" on page 476

Function	Description	More Information
Random Summary	Randomly displays the summary of a collection item	"Random Summary" on page 477
RSS Collection	Creates an RSS feed out of the items in a collection.	"RSS Collection Function" on page 469
Content Functions		
Add Content	Allows a user to add content to Ektron CMS400.NET programmatically.	"Add Content" on page 424
Content Block	Displays and manages a content block	"Content Block Function" on page 416
Form	Displays a content block associated with a form	"Form Function" on page 485
Multipurpose	Displays either a content block or a form	"MultiPurpose Function" on page 487
List Summary Functions		
Archive ListSummary	Displays summaries of archived content blocks	"Archive List Summary Function" on page 455
List Summary	Displays a list of content block summaries	"List Summary Function" on page 442
List Summary XML	Creates XML for a list summary by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML List Summary Function" on page 450
RSS List Summary	Creates an RSS feed out of the items in a list summary.	"RSS Summary Function" on page 460
Single Summary	Displays the summary for a content block	"Single Summary Function" on page 456
Login Functions		
Login	Adds the login or logout button	"Login Function" on page 414


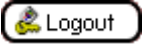
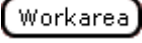
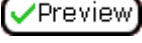

Function	Description	More Information
Menu Functions		
DropHTMLMenu	Used to display a DHTML menu	"DHTML Menus" on page 495
GenerateMenu	Works with DropHTML menu to display DHTML menus	"DHTML Menus" on page 495
XML Menu	Creates XML for a menu by wrapping tags around all of the menu items properties.	"XML Menu" on page 499
Metadata Functions		
Metadata	Displays the metadata for the current Web page	"Metadata Function" on page 488
Search Functions		
Search	Inserts a text box with a search button	"Search Functions" on page 428
Search Display	Creates an area to display search results	"Search Display Function" on page 438
Site Language Functions		
ShowSiteLanguage	Lets a site visitor select a language in which to view site.	"Show Site Language Function" on page 494

Login Function

The login function adds a login button on the template when displayed in a browser. Here is the format of the login function.

```
<% ecmLogin %>
```

This tag is responsible for displaying the following buttons on the Web page.

Button	Name	Description
	Login	When user is not logged into Ektron CMS400.NET, this button appears. Click this button to let the user log in to the application.
	Logout	Once logged in, this button replaces the login button to allow the user to log out from the Ektron CMS400.NET Web site.
	Workarea	When logged in, this button appears under the logout button allowing the user to access their workarea.
	Preview On	Lets the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off preview mode.

Login Tag Example

The following example places a login button in a table cell inside an asp template.

```
<tr>
  <td>
    <% ecmLogin %>
  </td>
</tr>
```

When inserted properly, a login button appears on the Web page.



Clicking the login button opens a login window in which a user enters the username and password. Upon successful completion, the user is logged in to the Ektron CMS400.NET Web site.

Placing Your Login Button

You can add as many login buttons to a template as you like. You might want to include a login button on each template, or have a special Web page called `login.asp`, from which users can log in without the public being able to access the login page.

Content Block Function

The content block tag displays a content block on an Ektron CMS400.NET Web page. There are two functions to choose from:

- `ecmContentBlock`
- `ecmContentBlockEX`

`ecmContentBlock` allows you display a content block without having to define an XSLT.

With `ecmContentBlockEX`, you can display XML content by defining an XSLT or applying one of the predefined XSLTs located in **Ektron CMS400.NET**. For more information on predefined XSLTs, see the **Administrator's Manual** > "Managing XML" > "Adding a Smart Form" > "Adding a Smart Form Using External XML Files".

Here is the format for the `ecmContentBlock` function.

```
<%ecmContentBlockEX ID %>
```

Here is the format for the `ecmContentBlockEX` function.

```
<%ecmContentBlockEX ID,
                        "XSLT",
                        Override %>
```

NOTE As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 487](#).

The table below explains the attributes of the `ecmContentBlock` function.

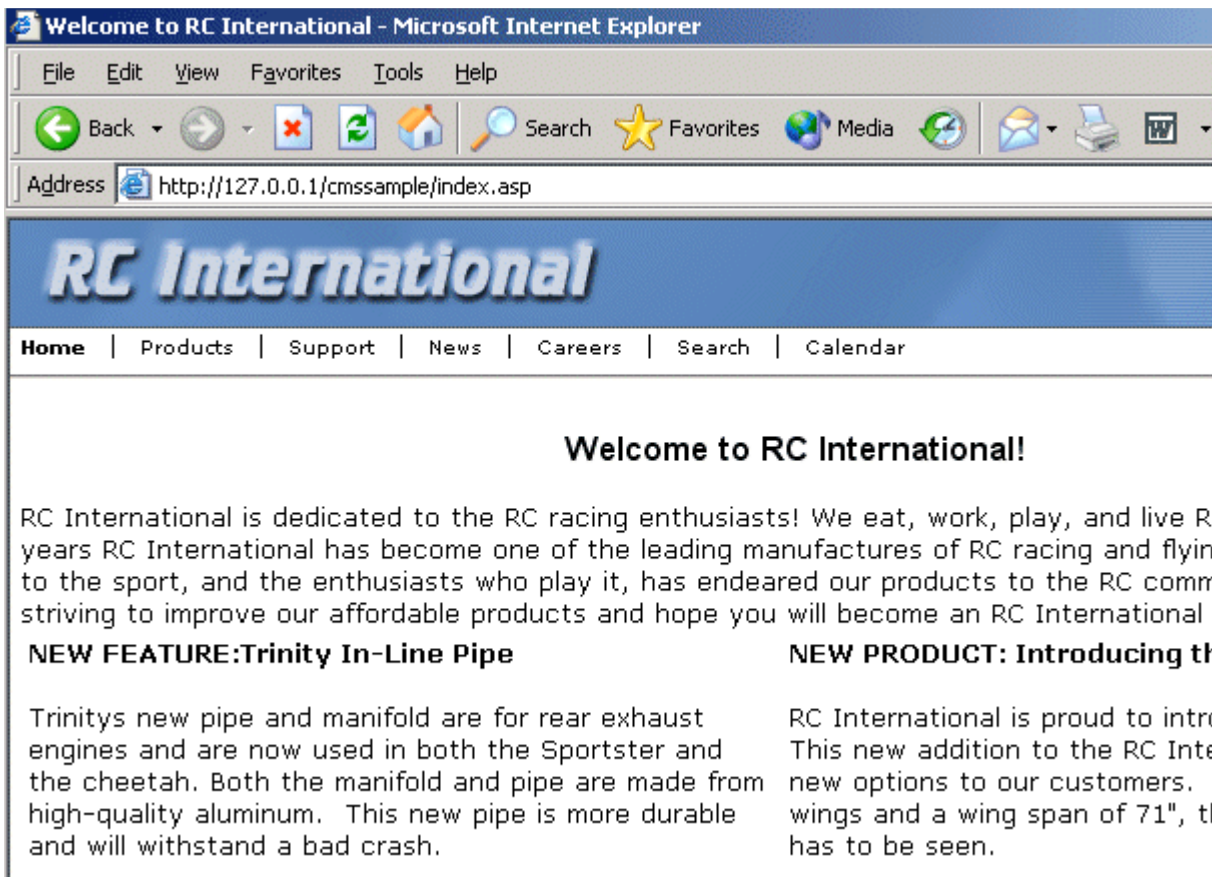
Attribute	Description	Required
ID	ID number assigned of the content block that will appear	Yes

The table below explains the attributes of the `ecmContentBlockEX` function.

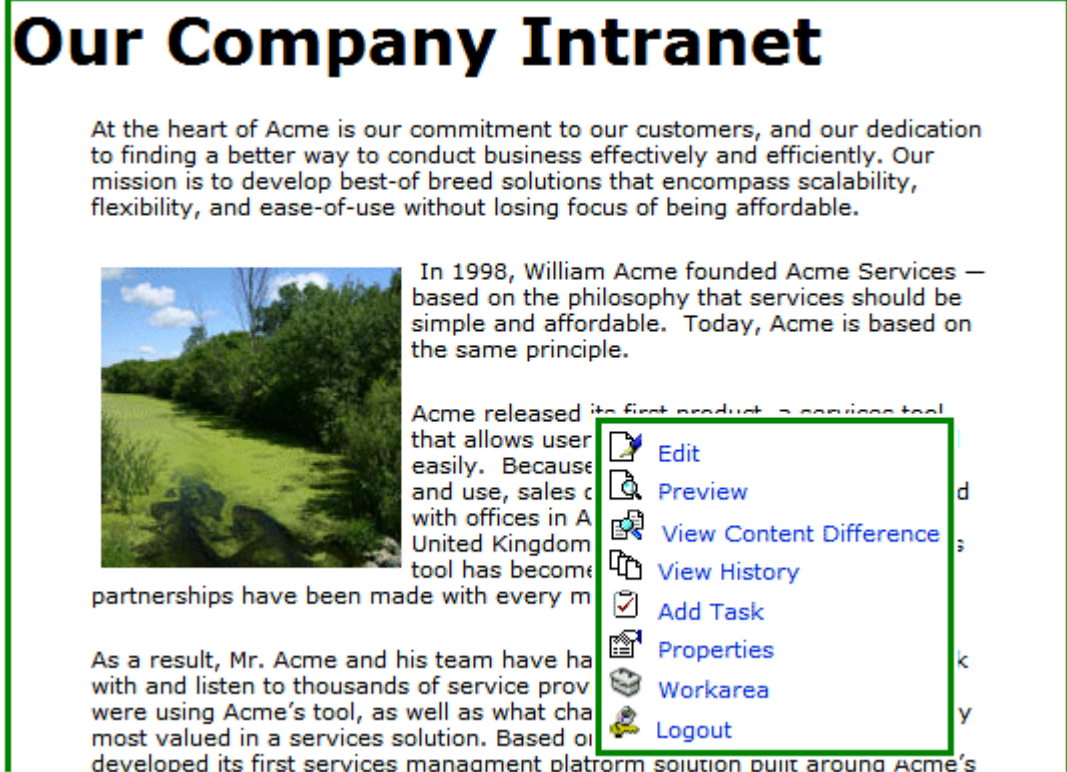
Attribute	Description	Required
ID	ID number assigned of the content block that will appear	Yes
XSLT	File name and path to a valid XSLT to transform the XML content. See Also: "The Function's Arguments" on page 421	No

Attribute	Description	Required
Override	If no XSLT is specified, then enter 1 to use XSLT1, 2 for XSLT2, or 3 for XSLT3. See Also: "The Function's Arguments" on page 421	No

In normal view, these tags retrieve the content block from the database and displays it in the template.



If a user is logged into Ektron CMS400.NET, the content block tag displays a border around the content block and when they right click on the content, a menu is provided for managing it.



Content Block Examples

Ektron CMS400.NET has two types of content blocks.

Type	Displays	More information
Static	One content block on the Web page	"Static Content Block" on page 419
Dynamic	A content block whose ID is passed through the URL	"Dynamic Content Block" on page 419

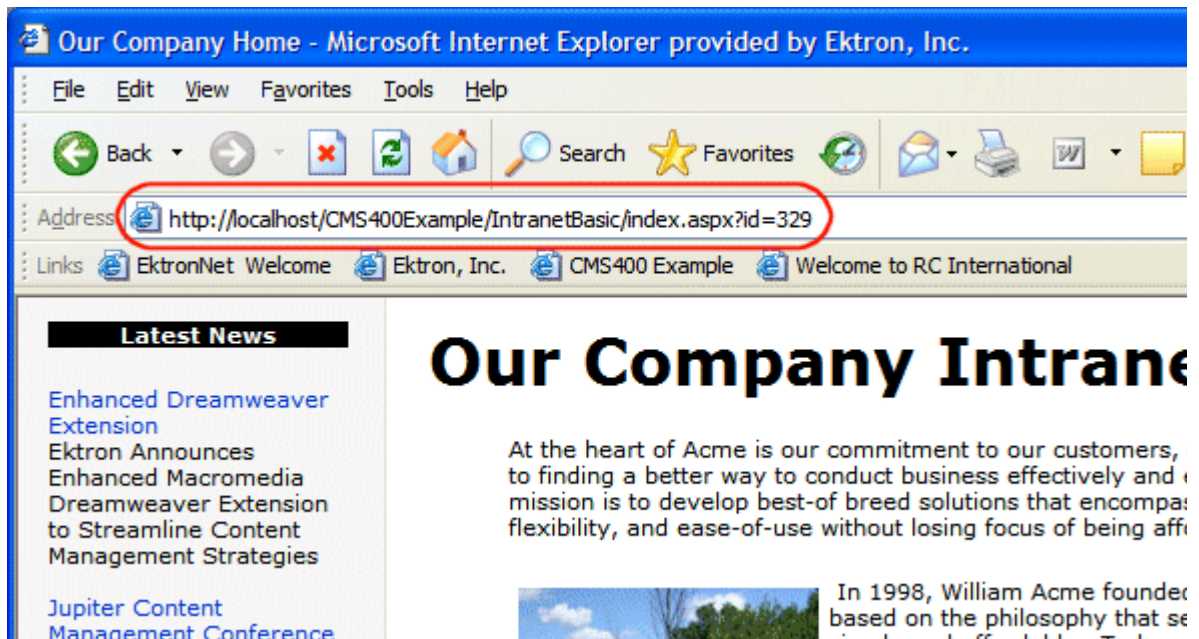
In addition, a content block can contain XHTML or XML content. For more information, see "XML Content Block" on page 421.

Static Content Block

The following illustrates how to insert a static content block on a template.

```
<tr>
  <td>
    <% ecmContentBlock(329) %>
  </td>
</tr>
```

In the example above, the function retrieves the content block with an id of 329 from the database to the browser.

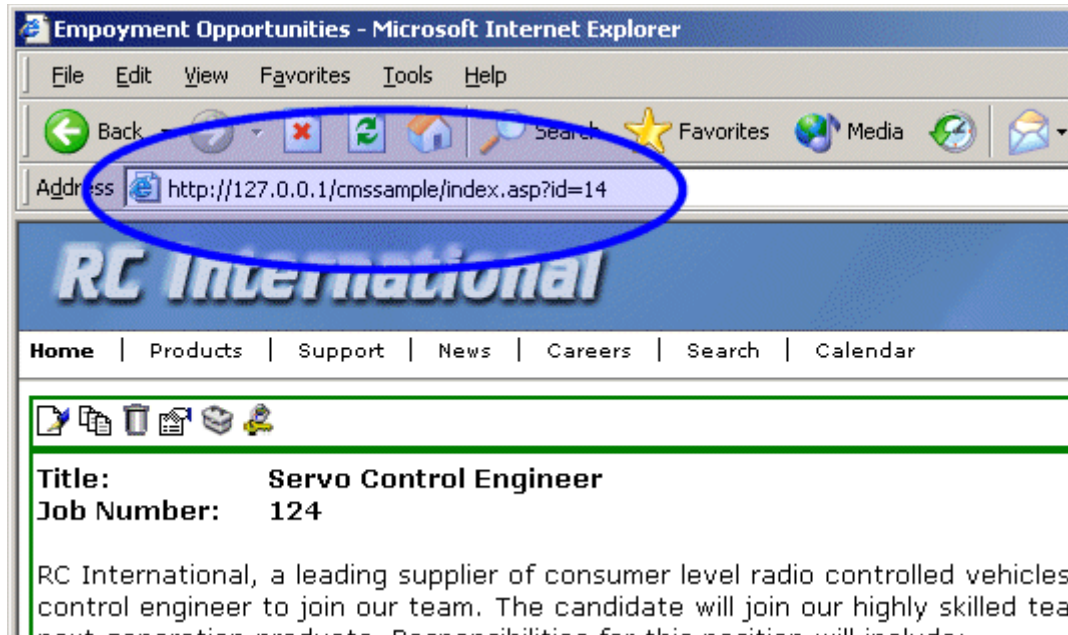


Dynamic Content Block

The following example shows the content block tag where the id is being passed as a URL parameter. Use this format with dynamic template.

```
<tr>
  <td>
    <% ecmContentBlock(request.QueryString("id")) %>
  </td>
</tr>
```

When a user accesses this page, it adds the content block ID number to the tag's template, and the content block with the corresponding ID appears on the Web page in the specified template.



The `id=` convention is used by **Ektron CMS400.NET** to generate quicklinks when content blocks are created. (For more information, see the **Ektron CMS400.NET** Administrator manual section "Managing Content" > "Forming URLs For Ektron CMS400.NET Documents.")

The dynamic `ecmContentBlock` function is almost the same as the static one (see "[Static Content Block](#)" on page 419). The only difference is the parameter, which makes the function dynamic. The parameter uses a query string call (`Request.QueryString`) to read the URL from the browser's address bar. It treats everything after the question mark as a list of key/value pairs separated by ampersands. So, `Request.QueryString` passes the key as a parameter and returns its associated value. For example, in URL `<http://localhost/cms400demo/index.asp?id=1&LangType=1036>`, the following `QueryString` calls return these values.

QueryString Call	Return value
<code>Request.QueryString("id")</code>	1
<code>Request.QueryString("LangType")</code>	1036

XML Content Block

Here is an example of displaying an XML content block. Note that you specify a content block and an XSLT, which determines how the content block is displayed.

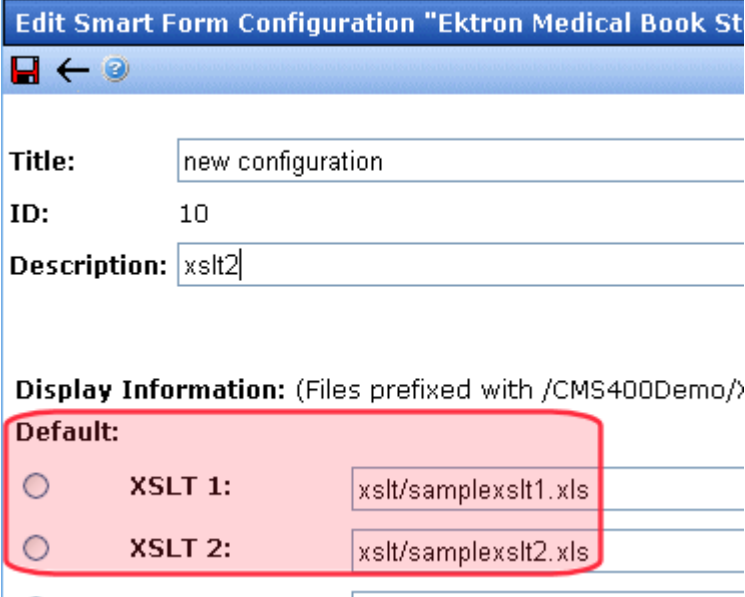
```
<tr>
  <td>
    <% ecmContentBlockEx 13, "", 1 %>
  </td>
</tr>
```

This example displays the content block ID=13, using XSLT1 as the display XSLT.

The Function's Arguments

The `ecmContentBlockEx` takes three arguments. The first argument identifies the content block to display. The second one is optional and, if used, specifies an external XSLT file.

If the second argument does not exist, the third argument specifies an XSLT identified in the Edit Smart Form screen. The following table provides additional detail about the second and third arguments.

To use this display XSLT	Enter this for argument 2	Enter this for argument 3
<p>The default XSLT specified in the Edit Smart Form screen (illustrated below). Note that, in this example, the XSLT Packaged option is the default XSLT since it is selected.</p> 	""	none
<p>XSLT Packaged: from the Edit Smart Form screen (that is, the XSLT created in the Data Designer)</p> <hr/> <p>Note: If an XSLT package has not been created, and zero (0) is the third argument, the default XSLT specified in the Edit Smart Form screen is used.</p> <hr/>	""	0
XSLT 1 from the Edit Smart Form screen	""	1
XSLT 2 from the Edit Smart Form screen	""	2
XSLT 3 from the Edit Smart Form screen	""	3

To use this display XSLT	Enter this for argument 2	Enter this for argument 3
An absolute or relative path to an XSLT	<p>An external XSLT file not specified in the Edit Smart Form screen.</p> <p>For example</p> <pre><% ecmContentBlockEx 13, "samplexslt. xsl"%></pre>	<p>does not matter</p> <p>- if a value exists in argument 2, argument 3 is ignored</p>

See Also: The "Managing XML" chapter of the **Ektron CMS400.NET** Administrator manual

Dynamic and Static Combination

Here is an example of how you can use the two different content tags in the same table cell by using an IF statement

```
<tr>
  <td>
    <% if request.QueryString("id")<> "" then
      ecmContentBlock(request.QueryString("id"))else
      ecmContentBlock(14) end if %>
  </td>
</tr>
```

In this sample code, when a person accesses this page without passing through the content's ID number, the content with an ID=14 will be displayed. If you pass a content id through the URL like **index.asp?id=4**, the content block ID=4 will be displayed.

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS400.NET without the need of logging in.

This function can be for several applications including, but not limited to:

- Applied to a Web form, allowing you to collect data from users about the content block
- Creating a loop function that moves information from a database to an Ektron CMS400.NET database

Here is the format for the add content function:

```
<% AddContent      Username,  
                   Password,  
                   Domain,  
                   content_title,  
                   content_comment,  
                   ContentHtml,  
                   SummaryHTML,  
                   ContentLanguage,  
                   FolderID,  
                   GoLive,  
                   EndDate,  
                   MetaInfo      %>
```

IMPORTANT! Every string that you pass to the function must be HTML encoded. Here is the VB.Net syntax for encoding a string.

```
myTitle = "John's Book"  
myTitle = Server.HtmlEncode(myTitle)
```

Explained in the table below are the attributes for the add content block function.

Attribute	Description
Username	The username of the user to be authenticated for permissions to see whether the user is able to add a content block.
Password	The password for the username being authenticated.
Domain	If using Active Directory, specify the domain needed to authenticate the user.
Content_title	Specify a title for the content block being added.

Attribute	Description
Content_comment	Specify a historical comment for the content being added.
ContentHTML	Actual content for the content block.
SummaryHTML	Summary information about the content block.
ContentLanguage	Language of content.
FolderID	ID number of the content folder where the content block will be added to.
GoLive	<p>Date and time the content will go live.</p> <hr/> <p>Must match date/time format set in the Web.config key <code>ek_dateFormat</code>.</p> <hr/>
EndDate	<p>Date and time the content will expire.</p> <hr/> <p>Must match date/time format set in the Web.config key <code>ek_dateFormat</code>.</p> <hr/>
MetalInfo	<p>Specify the information for the metadata which will be used as a reference point for finding other words or information.</p> <p>An example is:</p> <pre><metadata><meta id="3">Title</meta></metadata></pre> <p>Where ID is the metadata definition ID whose value is being set. The value is set between the meta tags and the meta tag is repeatable for as many metadata definitions as you want to set for the given content block.</p>

Add Content Example

The following is an example of the add content function in a template:.

```
<tr>
  <td>
    <% AddContent "admin", "admin", "", "Content Title", "Comment", "This is the
    <b>body</b> of the content.", "This is the <i>summary</i> for the content.", 1033, 31,
    "06-14-2004 11:55 AM", "", "<metadata><meta id="3">ContentTitle</meta></
    metadata>"," %>
  </td>
</tr>
```

The previous example has the following properties:

- The user “admin” is used to be authenticated.
- The title of the new content block is “Content Title”.
- The comment for the content is “Comment”.
- The body of the content is “This is the **body** of the content”.
- The summary of the content block is “This is the *summary* for the content.
- The language ID = 1033, or English.
- The folder ID = 31 is where the content is added.
- The start date is 06-14-2004”.
- No end date is specified.
- The metadata definition ID being set is “3” and the value is ContentTitle.

When this function is executed, the content is added to Ektron CMS400.NET.
Displayed here is the record of the content in the workarea.

View Content "Content Title"

View: English (U.S.) | Add: -select language-

Properties | Content | Summary | Meta Data | Alias | Comment | Tasks

Properties

Content Title:	Content Title
Content ID:	381
Content Language	English (U.S.)
Status:	Approved (Published)
Last User To Edit:	Application Administrator
Last Edit Date:	8/3/2005 12:27:47 PM
Start Date:	[None Specified]
End Date:	[None Specified]
Action on End Date	[None Specified]
Date Created:	8/3/2005 12:27:44 PM
Approval Method:	Do Not Force All Approvers
Approvals:	[None Specified]
XML Configuration:	[None Specified] HTML Content Assumed(inherited)
Path	\

Local intranet

Search Functions

The search tag is used to display a text box, with a search button on the template. When seen on the template of a Web page, users can enter text and click on the search button to search through all the HTML content on the Web site. To create a search, Ektron has created two search functions:

- ecmSearch
- ecmSearchNoFormTag

Both functions have the same parameters and perform the same search. The only difference is ecmSearch automatically inserts form tags, ecmSearchNoFormTag does not.

The following is an example of form tags.

```
<form name="ecmsearch" method="post" action='search-
display.asp'>
```

```
</form>
```

When using ecmSearch, form tags are added automatically. When using ecmSearchNoFormTag you need to add the tags manually

When using the search function, ecmSearch and ecmSearchNoFormTag can be interchanged. Shown here is the format for the ecmSearch and ecmSearchNoFormTag:

<pre><% ecmSearch or ecmSearchNoFormTag</pre>	<pre> "Starting Folder", Recursive, "Target Page", Text Box Size, Maximum Characters, "Button Image Source", "Button Text", "Font Face", "Font Color", "Font Size", Horizontal, "Spare" %></pre>
--	---

The following table provides information about all the attributes for the search tags.

Attribute	Description	Required ?	For more information, see
Starting Folder	The folder you wish to search.	Yes	"Starting Folder" on page 431
Recursive	Whether to search subfolders. 0=No, 1 = Yes	Yes	"Recursive" on page 431
Target Page	The Web page to output the search results.	Yes	"Target Page" on page 431
Text Box	The size of the text box for user input.	Yes	"Text Box" on page 432
Maximum Characters	The maximum number of characters that the input text box can accept.	No	"Maximum Characters" on page 432
Button Image Source	The image location for the search button. Blank ("") = standard submit button.	No	"Button Image Source" on page 433
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen.	No	"Button Text" on page 434
Font Face	The font of the search tag text.	No	"Font Face" on page 434
Font Color	The font color of the search tag text. Blank ("") = Black.	No	
Font Size	The font size of the search tag text.	No	"Font Size" on page 436
Horizontal	Determines if the search options are in a row or a column. 1 = horizontal, 0 = vertical	Yes	"Horizontal" on page 436
Spare1	Currently not used.	Must be ""	"Spare 1" on page 437

Search Tag Example

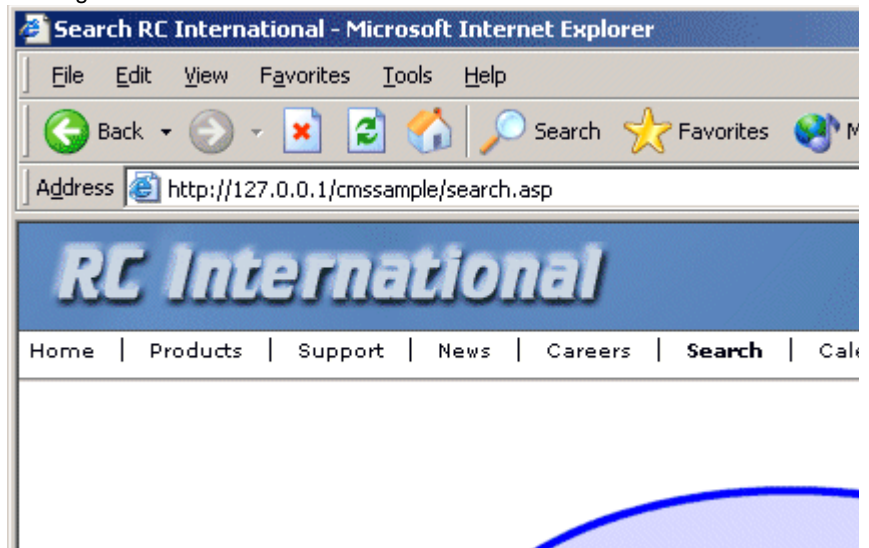
The following example places a text field and a search button in a table cell inside an ASP template.

```
<tr>
  <td>
    <% ecmSearch "\",1,"search-
      display.asp",25,200,"","Search","Verdana",
      "#808080","2",0,"" %>
  </td>
</tr>
```

The previous example has the following properties:

- Searches the root folder, in this case the \content folder.
- Search is recursive.
- The target page is "search-display.asp".
- The size of the text box is 25.
- The search text box allows up to 200 characters.
- No image is used for a button. Instead, a standard submit button is used.
- The text on the submit button is "Search".
- The search options appear in Verdana with a color of #808080 and a size of 2pt.
- The search options are arranged vertically.

When added to a template, and visited, the search tag will appear similar to the following:



Search Tag Attributes

The search tag's attributes are described below.

Starting Folder

In this attribute, specify the content folder where the search begins. There are two options for this attribute.

Value	Search begins in
"\"	root folder
"\folder path"	specified folder

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this attribute:

Value	Search begins in
1	Recursive search (will search sub-folders of the starting folder).
0	Non-Recursive search (will only search starting folder).

Target Page

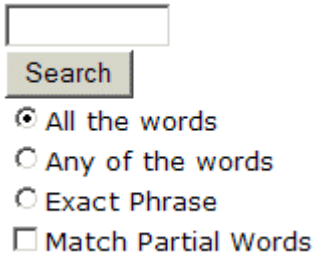
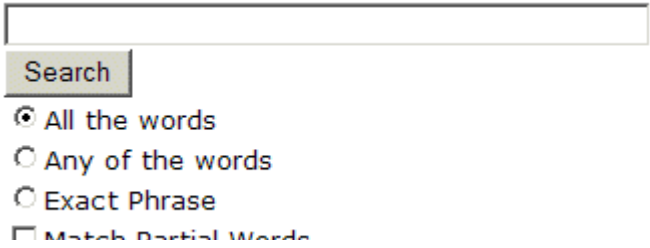
The page that displays the search results. You may enter any form of a URL for this page.

NOTE The target page *must* include the `searchdisplay` function to display search results.

To use the same page, enter a pound sign (#) into the field.

Text Box

This attribute sets the size of the search text box. It can be any number, depending on the size of the search text box that you want.

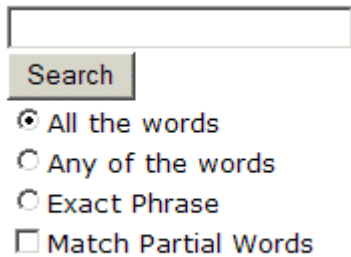
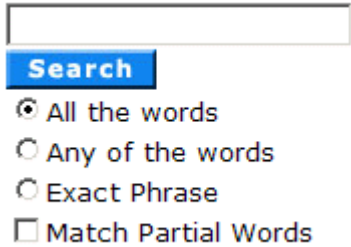
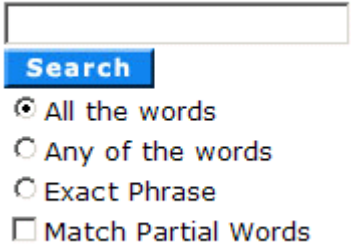
Text Box Size	Web Page View
10	
50	

Maximum Characters

This attribute determines the maximum number of characters that a user can enter for search text.

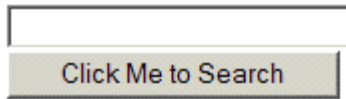
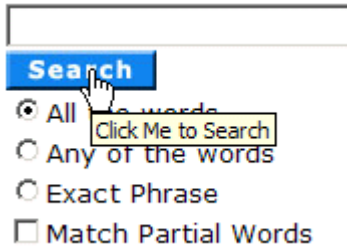
Button Image Source

To use an image for the search button, enter the image's filename and path.

Button Image Source	Web Page View
""	 <p>A screenshot of a web page search interface. It features a text input field at the top. Below the input field is a search button with a light gray background and the word "Search" in black text. Underneath the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (which is a checkbox option).</p>
"search.jpg"	 <p>A screenshot of a web page search interface. It features a text input field at the top. Below the input field is a search button with a blue background and the word "Search" in white text. Underneath the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (which is a checkbox option).</p>
"http://www.ektron.com/images/search.jpg"	 <p>A screenshot of a web page search interface. It features a text input field at the top. Below the input field is a search button with a blue background and the word "Search" in white text. Underneath the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (which is a checkbox option).</p>

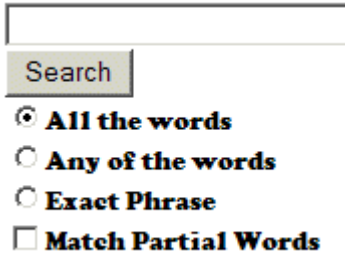
Button Text

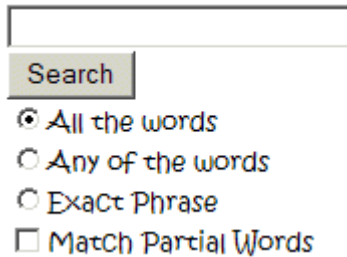
If no search button image is specified, the text in this field appears on the button.
If an image is being used for the search button, this text is the image's alt text.

Button Text	Web Page View
"Click Me to Search"	 <p> <input checked="" type="radio"/> All the words <input type="radio"/> Any of the words <input type="radio"/> Exact Phrase <input type="checkbox"/> Match Partial Words </p>
"Click Me to Search"	 <p> <input checked="" type="radio"/> All the words <input type="radio"/> Any of the words <input type="radio"/> Exact Phrase <input type="checkbox"/> Match Partial Words </p>

Font Face

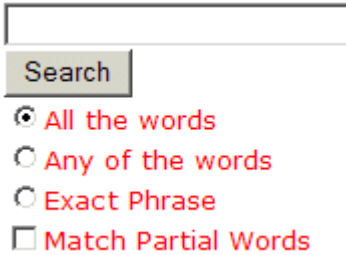
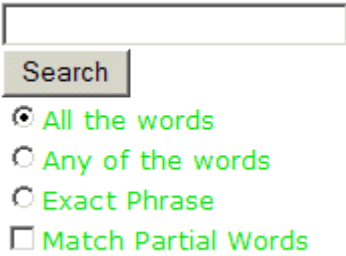
This font is used for the search options displayed on the Web page.

Font Face	Web Page View
Cooper Black	 <p> <input checked="" type="radio"/> All the words <input type="radio"/> Any of the words <input type="radio"/> Exact Phrase <input type="checkbox"/> Match Partial Words </p>

Font Face	Web Page View
Kristen ITC	 <p>The screenshot shows a search interface with a text input field, a 'Search' button, and four radio button options: 'All the words', 'Any of the words', 'Exact Phrase', and 'Match Partial Words'. The text in the interface is rendered in the 'Kristen ITC' font.</p>

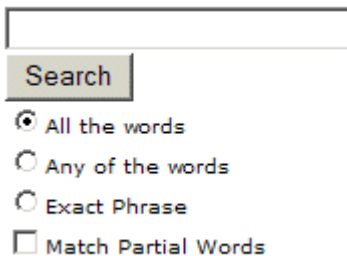
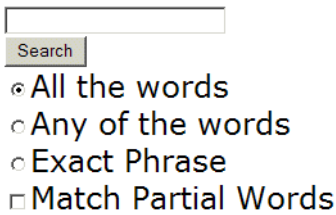
Font Color

The color of the text in the search tag. You may enter the color name or a hexadecimal value of the color.

Font Color	Web Page View
Red	 <p>The screenshot shows the same search interface as above, but the text is rendered in red. The options are: 'All the words', 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>
#14DA14	 <p>The screenshot shows the same search interface as above, but the text is rendered in green. The options are: 'All the words', 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>

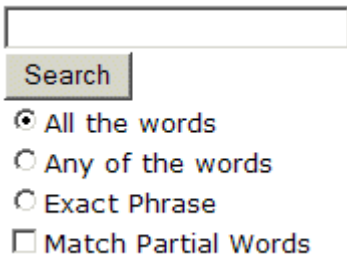
Font Size


This is the size of the text on the search tag.

Font Size	Web Page View
1	 <p>A search tag with a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>
5	 <p>A search tag with a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox). The text is larger than in the font size 1 example.</p>

Horizontal

This attribute determines whether the search options are displayed vertically or horizontally.

Attribute	Web Page View
0 (vertical)	 <p>A search tag with a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox). The options are displayed vertically.</p>

Attribute	Web Page View
1 (horizontal)	

Spare 1

At the end of the Search tag is a attribute called Spare 1. This attribute has not yet been assigned an function. As of now, when you enter a search tag, you must pass this attribute "".

Search Display Function

The search display custom tag creates the area where search results appear after a search is executed.

Here is the format of the ecmSearchDisplay tag.

```
<% ecmSearchDisplay          "MaxNumberReturn",
                             "StyleInfo",
                             ShowDate          %>
```

Below is a description of the tag's attributes.

Attribute	Description	Required	For more information, see
Max Number of content blocks Returned	Maximum number of content blocks returned. (0=Unlimited)	Yes	"Max Number of Content Blocks Returned" on page 440
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc.	No	"Style Info" on page 441
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes	No	"Show Date" on page 441

Search Display Example

Insert the following on the page that displays search results. You may place it on the same page as the search tag or a separate page.

```
<tr>
  <td>
    <% ecmSearchDisplay 0,"","1" %>
  </td>
</tr>
```

This search display example has the following properties:

- Returns unlimited results when the search is performed.
- Contains no style information.
- Shows the last modified date of the content.

When the tag is added to a template, and a search is performed, the results appear similar to the following.



Search Display with Default Content

You can also set up the search display tag to display a default content block if a search is not performed. Then, if a search is performed, the results replace the content block. The sample code below does that:

```
<tr>
  <% if (Request.Form("ecmsearchtype") <> "") then %>
    <td>
      <% ecmSearchDisplay 0,"" %>
    </td>
  <% else %>
    <td>
      <% ecmContentBlock(1) %>
    </td>
  <% end if %>
</tr>
```

Search Display Attributes

Listed below are the attributes for the SearchDisplay tag.

Max Number of Content Blocks Returned

This attribute determines the number of search results displayed on a page. You may enter any integer. To display unlimited results, enter 0 (zero).

Max Returns	Web Page View
0	Search Results RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah
3	Search Results RC Sportster RC Redstar RC Lilly

Style Info

An HTML style string used within the HTML “hyperlink” generated by the search output. Can control font, color, size, etc.

Style Info	Web Page View
""	<p>Search Results</p> <p>RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah</p>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"	<p>Search Results</p> <p>RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah</p>

Show Date

This attribute determines if the content block's last modified date appears next to the title. This is useful when determining which content block is the most recent. You have two choices.

Attribute	Web Page View
0	Search Results RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah
1	Search Results RC Sportster (8/27/2002 7:52:40 PM) RC Redstar (8/27/2002 7:52:01 PM) RC Lilly (8/27/2002 7:51:26 PM) RC Cheetah (8/27/2002 7:50:53 PM) Products Page (8/27/2002 7:50:09 PM) RC International Ships RC Sportster (8/27/2002 2:46:12 PM) RC International Hosts RC Conference (8/27/2002 2:44:44 PM) RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM)

List Summary Function

The list summary function is used to display the list of content summaries on a Web page.

Shown here is the format of the list summary function:

```
<% ecmListSummary      "Folder",
                        Recursive,
                        ShowSummary,
                        "StyleInfo",
                        "OrderBy, Asc/Desc",
                        "ShowInfo",
                        MaxNumber,
                        "ObjType",
                        "OptionList"      %>
```

A list and description of the attributes are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
Recursive	Allow the tag to get summary info from the child folders.	0 = Not Recursive 1 = Recursive
ShowSummary	Do you want the content summary to be displayed?	0 = No 1 = Yes
StyleInfo	Inline style information for the generated text hyperlink.	Any string
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "Startdate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
ShowInfo	Information that will appear to the right of the hyperlink.	"DateModified" "DateCreated" "AuthorFname" "AuthorLname"
MaxNumber	Maximum number of summaries returned.	0 = Unlimited

Attribute	Description	Options
ObjType	<p>Determines whether the list summary is for the folder or a single piece of content.</p> <p>For example:</p> <p>If the folder property = \books and "Folder" is chosen as the ObjType, the list summary for the folder with books appears.</p> <p>If the folder property = 1 and you choose "Content" as the ObjType, a single summary for content with an id = 1 appears.</p> <p>See Also: "Single Summary Function" on page 456</p> <p>The default ObjType is "Folder" when left blank.</p>	<p>""</p> <p>"Folder"</p> <p>"Content"</p>
OptionList	Not yet implemented	MUST BE ""

List Summary Example

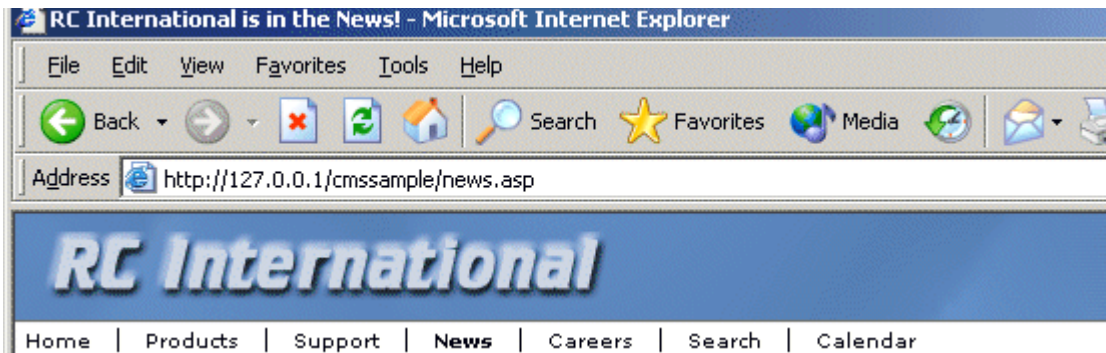
Shown below is an example of the ListSummary function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmListSummary "\Marketing\News",0,1,"",
      "Title,asc","DateModified",0,"","" %>
  </TD>
</TR>
```

The previous example has the following properties:

- Displays summaries from the \Marketing\News folder.
- Not recursive.
- The summaries will be shown.
- No style was applied to the hyperlinks.
- The results will be displayed in ascending order by title.
- The date modified of each content block will be displayed.
- Unlimited results will be shown.

Here is how the summaries will appear on the web page.



List Summary Attributes

Listed below are all the different attributes and descriptions about them.

Starting Folder

In this attribute, you will need to specify which content folder you will want to get all the summaries from. There are two options for this attribute:

Value	Description
""	By leaving the value blank, all the summaries from the root content folder will be displayed.
"Folder Path"	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the subfolders associated with the starting folder you had specified. You have two options for this attribute:

Value	Description
1	Recursive - summaries in the starting folder's subfolders will also be displayed.
0	Non-recursive - only summaries in the starting folder will be displayed.

Show Summary

The show summary attribute displays or suppresses the summary in the summary list. You have two options for this attribute.

Attribute	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>RC International Hosts RC Conference</p> <p>RC International Announces the RC Cheet</p>
1	<p>RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM) RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p> <p>RC International Hosts RC Conference (8/27/2002 2:44:44 PM) RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p>

Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	<p>RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM) RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p> <p>RC International Hosts RC Conference (8/27/2002 2:44:44 PM) RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p>

Style Info	Web Page View
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;"	<p>RC International Announces the RC Cheetah</p> <p>RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p> <p>RC International Hosts RC Conference</p> <p>RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference Atlanta, Georgia.</p>

Order By

This attribute will sort the summary results by different options. There are five different ways to sort the summaries:

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
Startdate	The Start Date assigned to the content block when created.
AuthorFname	author's first names
AuthorLname	author's last name

Ascending/Descending

This attribute is an addition to the previous OrderBy attribute. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to rearrange these to display either in ascending or descending order.

To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order

Value	Description
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are four different options that you may choose from:

Value	Description	Web Page View
DateModified	Displays the date the content was last modified.	, RC Sportster. The RC Sportster is design mid level RC racer currently on the market ference (8/27/2002 2:44:44 PM) growing company in the RC host the 3rd annual RC conference in Atl
***DateCreated	Displays the date when the content was first created.	1 level RC racer currently on the mark ference (4/25/2002 4:42:55 PM) owing company in the RC ost the 3rd annual RC conference in A
AuthorFname	Displays the first name of the user who last edited the content block.	mid level RC racer currentl ference (John) growing company in the R host the 3rd annual RC cc

Value	Description	Web Page View
AuthorLname	Displays the last name of the user who last edited the content block.	ct, KC Sportster. The KC Sp d mid level RC racer currentl Conference (Doe) st growing company in the R to host the 3rd annual RC c

Max Number

This attribute determines how many summaries appear on a page. Enter the number of summaries to be displayed. To display all summaries, enter 0 (zero) and you will display unlimited results.

XML List Summary Function

The XML List Summary function wraps XML around the properties of a list of content that's created as a list summary in the Ektron CMS400.NET Workarea. Once created, you can use the XML to display the list summary information. The `ecmListSummaryXML` function allows you to easily define how the list summary data appears on the Web page.

Shown here is the format of the `ecmListSummaryXML` function.

```
<% ecmListSummaryXML      "Folder",  
                           "Recursive",  
                           "OrderBy",  
                           "MaxNumber",  
                           "ObjType",  
                           "SummaryType"      %>
```

The `ecmListSummaryXml` function attributes are explained below.

Attribute	Description	Options
Folder	This is the folder the summary information will come from. <i>See Also:</i> "Starting Folder" on page 445	Any folder
Recursive	Allow the tag to get summary info from the child folders. <i>See Also:</i> "Recursive" on page 445	0 = Not Recursive 1 = Recursive
OrderBy, ASC/ DESC	Order the hyperlinks by: <i>See Also:</i> "Order By" on page 447	"Title" "DateModified" "DateCreated" "Startdate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
MaxNumber	Maximum number of summaries returned. <i>See Also:</i> "Max Number" on page 449	0 (Zero) = Unlimited

Attribute	Description	Options
ObjType	<p>Determines whether the list summary is for the folder or a single piece of content.</p> <p>For example:</p> <p>If the folder property = \books and "Folder" is chosen as the ObjType, the list summary for the folder with books appears.</p> <p>If the folder property = 1 and you choose "Content" as the ObjType, a single summary for content with an id = 1 appears.</p> <p>See Also: "Single Summary Function" on page 456</p> <p>The default ObjType is "Folder" when left blank.</p>	<p>""</p> <p>"Folder"</p> <p>"Content"</p>
SummaryType	<p>Used as a filter, this property determines the type of content that appears in the list summary. The default is "Content" when left blank.</p>	<p>"AllTypes" - displays all of the content types for the given folder.</p> <p>"Content" - displays a list of content blocks.</p> <p>"Forms" - forms appear in the list summary.</p> <p>"Archive_Content" - archived content blocks appear in the list summary.</p> <p>"Archive_Forms" - archived forms appear in the list summary.</p> <p>"Assets" - assets, such as offices documents, appear in the list summary.</p> <p>"Archive_Assets" - archived assets appear in the list summary.</p> <p>"LibraryItem" - library items appear in the list summary.</p> <p>"NonLibraryContent" - all types of content appear in the list summary except for library items.</p>

When using the XML list summary function, the following tags are created around the properties of each item in the list summary.

XML Tags	Description
<code><Content></Content></code>	Declares a new section of content in the XML collection.
<code><ID></ID></code>	The content block ID. For example: <code><ID>27</ID></code>
<code><Title></Title></code>	The title of the content block. For example: <code><Title>HTML for the World Wide Web with XHTML and CSS</Title></code>
<code><QuickLink></QuickLink></code>	The quicklink associated with the content block. For example: <code><QuickLink>/cms400demo/collection.aspx?id=27</QuickLink></code>
<code><Teaser></Teaser></code>	The summary of the content block. For example: <code><Teaser>The easiest HTML book weve seen that still manages to be comprehensive.</Teaser></code>
<code><StartDate></StartDate></code>	The content block's start date formatted as a .NET date type For example: <code><StartDate>1/1/0001 12:00:00 AM</StartDate></code>
<code><DateModified></DateModified></code>	The last date the content block was modified. For example: <code><DateModified>1/19/2005 4:18:06 PM</DateModified></code>
<code><EndDate></EndDate></code>	The date the content block stops running on the Web site For example: <code><EndDate>12/31/9999 12:00:00 AM</EndDate></code>
<code><LastEditorFname></LastEditorFname></code>	The first name of the last person who edited the content block. For example: <code><LastEditorFname>John</LastEditorFname></code>

XML Tags	Description
<code><LastEditorLname></LastEditorLname></code>	The last name of the last person who edited the content block. For example: <code><LastEditorLname>Johnson</LastEditorLname></code>
<code><DisplayStartDate></DisplayStartDate></code>	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.
<code><FolderID></FolderID></code>	The ID of the folder in which the content block is contained. For example: <code><FolderID>0</FolderID></code>
<code><ContentStatus></ContentStatus></code>	The status of the content block.
<code><Language></Language></code>	The language of the content block. For example: <code><Language>1033</Language></code>
<code><DisplayDateModified></DisplayDateModified></code>	edited. It is formatted as a string that represents Ektron CMS400's display of the date. For example: <code><DisplayDateModified>1/19/2005 4:18:06 PM</DisplayDateModified></code>
<code><DisplayEndDate></DisplayEndDate></code>	The content block's end date. It is formatted as a string that represents Ektron CMS400's display of the date. For example: <code><DisplayEndDate>12/31/9999 12:00:00 AM</DisplayEndDate></code>

Here is how the raw XML list summary output appears before an XSLT is applied.

```
<Content>
  <ID>24</ID>
  <Title>ASP.NET Unleashed</Title>
  <QuickLink>/cms400demo/collection.aspx?id=24</QuickLink>
  <Teaser><p><em>ASP.NET Unleashed, Second Edition</em> is really big, really thorough, and
really <i>good</i>.</p></Teaser>
  <StartDate>1/1/0001 12:00:00 AM</StartDate>
  <DateModified>1/12/2005 7:53:31 PM</DateModified>
  <EndDate>12/31/9999 12:00:00 AM</EndDate>
  <LastEditorFname>Application</LastEditorFname>
  <LastEditorLname>Administrator</LastEditorLname>
  <DisplayStartDate></DisplayStartDate>
  <FolderID>0</FolderID>
  <ContentStatus></ContentStatus>
  <Language>0</Language>
  <DisplayDateModified>1/12/2005 7:53:31 PM</DisplayDateModified>
  <DisplayEndDate></DisplayEndDate>
</Content>
<Content>
  <ID>25</ID>
  <Title>Microsoft ASP.NET Programming with Microsoft Visual C#.NET</Title>
  <QuickLink>/cms400demo/collection.aspx?id=25</QuickLink>
  <Teaser><p>Teach yourself how to write high-performance Web applications with ASP.NET and
Visual C# .NET--one step at a time.</p></Teaser>
  <StartDate>1/1/0001 12:00:00 AM</StartDate>
  <DateModified>1/4/2005 7:16:21 PM</DateModified>
  <EndDate>12/31/9999 12:00:00 AM</EndDate>
  <LastEditorFname>Application</LastEditorFname>
  <LastEditorLname>Administrator</LastEditorLname>
  <DisplayStartDate></DisplayStartDate>
  <FolderID>0</FolderID>
  <ContentStatus></ContentStatus>
  <Language>0</Language>
  <DisplayDateModified>1/4/2005 7:16:21 PM</DisplayDateModified>
  <DisplayEndDate></DisplayEndDate>
</Content>
```

Archive List Summary Function

The Archive list summary function displays on a Web page a list of content summaries for archived content blocks. For example, you might have a current news page and an archived news page. Use the List Summary Function to display current news, and this function to display older news stories.

On the Content Block **schedule** screen, the user must set an **End Date** then choose the Archive and Display option to have the content blocks appear via this function.

Shown here is the format of the Archive List Summary function:

```
<% ecmListArchiveSummary      "StartingFolder",  
                               Recursive,  
                               ShowSummary,  
                               "StyleInfo",  
                               "OrderBy, Asc/Desc",  
                               "ShowInfo",  
                               MaxNumber,  
                               "Spare1",  
                               "Spare2"  
                               %>
```

This function has the same parameters as the List Summary Function. See ["List Summary Function" on page 442](#) for details.

Single Summary Function

The single summary tag is used to display the summary of one content block in the Ektron CMS400.NET Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<% ecmSingleSummary      "ContentID",
                          ShowSummary,
                          "StyleInfo",
                          "ShowInfo",
                          "Spare1"      %>
```

You can look at the single summary tag as just a simpler form of the List summary tag. The format for each of them is almost identical, but the single summary tag does not require as many attributes to be passed.

A list and description of the attributes are shown in this table:

Attribute	Description	Required?
Content ID	This is the ID number of the content block the summary information will come from.	Yes
ShowSummary	Do you want the content summary to be displayed?	Yes
StyleInfo	Inline style information for the generated text hyperlink.	No
ShowInfo	Information that will appear to the right of the hyperlink.	No
Spare1	Not yet implemented	MUST BE ""

Single Summary Example

Shown below is an example of the single summary function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmSingleSummary "14",1,"","DateModified","" %>
  </TD>
</TR>
```

The previous example had the following properties:

- Displays the summary for the content block with an ID=14

- The summary will be shown
- No style was applied to the hyperlinks
- The last date the content was modified will be displayed

Here is how the summary will appear on the web page.

Servo Control Engineer #124 (10/29/2002 6:51:02 PM)
RC International is looked for a highly skilled servo control engineer to join our team.

Single Summary Attributes

Listed below are all the different attributes and descriptions about them.

Content ID

In this attribute, you will need to specify which content block you will want the summary to be displayed from.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this attribute:

Value	Web Page View
0	Servo Control Engineer #124
1	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Style Info

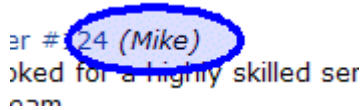
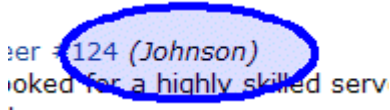
In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.
font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;	<div style="border: 1px solid blue; background-color: #cccccc; padding: 2px; display: inline-block;"> Servo Control Engineer #124 </div> RC International is looked for a highly skilled servo control engineer to join our team.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are five different options that you may choose from:

Value	Displays the...	Web Page View
ID	ID number for the content block.	Servo Control Engineer #124 (14) RC International is looked for a highly skilled servo control engineer to join our team.
DateModified	date the content was last modified.	Servo Control Engineer #124 (8/27/2002 2:40:26 PM) RC International is looked for a highly skilled servo control engineer to join our team.
DateCreated	date when the content was first created.	Servo Control Engineer #124 (4/25/2002 4:54:42 PM) RC International is looked for a highly skilled servo control engineer to join our team.

Value	Displays the...	Web Page View
AuthorFname	first name of the user who last edited the content block.	
AuthorLname	last name of the user who last edited the content block	

Spare 1

There is one attribute fields at the end that are currently not used. This is here for future scalability. When you insert a single summary tag into your template, you must leave this attributes as ""

RSS Summary Function

The RSS summary function is used to create an XML list of content summaries formatted for RSS Syndication.

Shown here is the format of the RSS summary function:

```
<% ecmRssSummary      "Folder",
                        Recursive,
                        "OrderBy, Asc/Desc",
                        MaxNumber,
                        %>
```

A list and description of the attributes are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
Recursive	Allow the tag to get summary info from the child folders.	0 = Not Recursive 1 = Recursive
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "Startdate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
MaxNumber	Maximum number of summaries returned.	0 = Unlimited

RSS Summary Example

Shown below is an example of the ecmRssSummary function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmRssSummary "\CMS400Demo\", 0, "Title,asc", 0 %>
  </TD>
</TR>
```

The previous example has the following properties:

- Displays summaries from the \CMS400Demo\ folder.
- Not recursive.
- The results will be displayed in ascending order by title.
- Unlimited results will be shown.

Here is how an RSS summary appears when viewed without an RSS reader on a web page.

```
<?xml version="1.0" ?>
- <rss xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" version="2.0">
- <channel>
  <title>example listsummary</title>
  <link>http://localhost/CMS400Demo/developer/ListSummary/RSS_ListSummary.aspx?show=true</link>
  <description />
- <item>
  <title>eWebEditPro+XML V4.2 supports the vision of create content once, reuse many times</title>
  <link>http://192.168.0.82/CMS400Demo/Developer/ListSummary/pr.aspx?id=290</link>
- <description>
- <![CDATA[ &lt;p&gt;Ektrons XML Authoring Tool Makes it Easier to Create Structured, Validated Content to
Support an Organizations Information and Content Management Strategies&lt;/p&gt; ]]>
  </description>
<author>Application Administrator</author>
<pubDate>3/7/2006 2:00:46 PM</pubDate>
</item>
- <item>
  <title>Ektron Offers a Visual Development Environment for Rapid CMS Integration and Deployment</title>
  <link>http://192.168.0.82/CMS400Demo/Developer/ListSummary/pr.aspx?id=282</link>
- <description>
- <![CDATA[ &lt;p&gt;Ektron leverages strong capabilities of Visual Studio.NET to give developers a truly visual
environment for integrating CMS components into Web templates&lt;/p&gt; ]]>
  </description>
  <author>Application Administrator</author>
  <pubDate>3/7/2006 2:02:19 PM</pubDate>
  </item>
- <item>
  <title>Ektron Announces Winner of All-Stars Customer Competition</title>
  <link>http://192.168.0.82/CMS400Demo/Developer/ListSummary/pr.aspx?id=292</link>
- <description>
- <![CDATA[ &lt;p&gt;&lt;em&gt;-- All-Stars competition identifies the most innovative and effective
implementations of Ektrons Web solutions and tools --&lt;/em&gt;&lt;/p&gt; ]]>
  </description>
  <author>Application Administrator</author>
  <pubDate>3/6/2006 10:51:52 AM</pubDate>
  </item>
</channel>
</rss>
```

RSS Summary Attributes

Listed below are all the different attributes and descriptions about them.

Starting Folder

In this attribute, you will need to specify which content folder you will want to get all the summaries from. There are two options for this attribute:

Value	Description
""	By leaving the value blank, all the summaries from the root content folder will be displayed.
"\Folder Path"	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the subfolders associated with the starting folder you had specified. You have two options for this attribute:

Value	Description
1	Recursive - summaries in the starting folder's subfolders will also be displayed.
0	Non-recursive - only summaries in the starting folder will be displayed.

Order By

This attribute will sort the summary results by different options. There are five different ways to sort the summaries:

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
Startdate	The Start Date assigned to the content block when created.
AuthorFname	author's first names

Value	Summaries sorted by
AuthorLname	author's last name

Ascending/Descending

This attribute is an addition to the previous OrderBy attribute. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to rearrange these to display either in ascending or descending order.

To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Max Number

This attribute determines how many summaries appear on a page. Enter the number of summaries to be displayed. To display all summaries, enter 0 (zero) and you will display unlimited results.

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS400.NET Workarea. The `ecmCollection` function is highly customizable, allowing you to easily define how the collection data appears on the Web page.

Shown here is the format of the `ecmCollection` function.

```
<% ecmCollection      "id",
                        "Display Function"      %>
```

The `ecmCollection` function attributes are explained below.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
displayFunction	Specify a function that is defined in the API and determines how the appearance of the collection data on a Web page. The display function must also be defined in the <code>ecmCollection</code> function.

See Also: The **Ektron CMS400.NET** User Manual Section "Working with Collections" > "Comparison of Collections, Menus, and the List Summary Features"

Collection Examples

The following are examples of the collection function in the Ektron CMS400.NET sample Web site. One example uses the `ecmNavigation` display function, and the other uses the `ecmTeaser` display function.

The `ecmNavigation` and `ecmTeaser` display functions are sample display functions included with Ektron CMS400.NET.

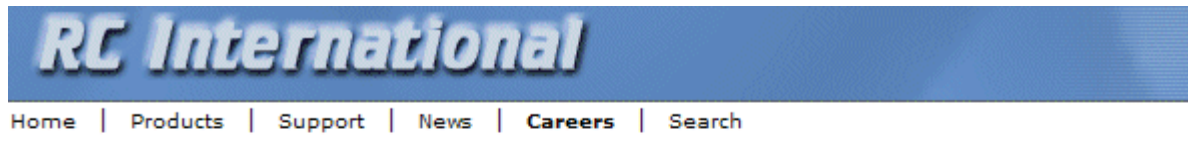
The `ecmNavigation` Display Function

The following sample collection function is being used as a navigation menu in the Ektron CMS400.NET sample Web site.

```
<% ecmCollection 1, "ecmNavigation" %>
```

In this example, the collection with an ID=1 is displayed, and the `ecmNavigation` function defines the appearance of the collection data on a Web page.

In this example, the collection with an ID=2 is displayed, and the `ecmTeaser` function defines the appearance of the collection data on the Web page. `ecmTeaser` displays the title and summary of each content block in the collection. Here is how the collection might appear on a Web page.



Employment Opportunities



Plastic Molder #123 (8/27/2002 2:38:34 PM)

RC International is looking for an experienced plastics molder.

Servo Control Engineer #124 (8/27/2002 2:40:26 PM)

RC International is looking for a highly skilled servo control engineer to join our team.

The format of the collection on the Web page depends on the displayFunction that is used. Below is the source code for the display function `ecmTeaser`.

```
Function ecmTeaser(cInfo)
    dim html, info
    html = "<table border=""0"" cellspacing=""0"" cellpadding=""0""
           width=""100%"">"
    for each info in cInfo
        html = html & "<tr><td><a href=""
        html = html & CStr(info("ContentLinks"))
        html = html & "" "" >"
        html = html & Cstr(info("ContentTitle"))
        html = html & "</a>&nbsp;("
        html = html & info("DateModified")
        html = html & ")</td></tr><tr><td>"
        html = html & Cstr(info("ContentTeaser"))
        html = html & "</td></tr><tr><td>&nbsp;</td></tr>"
    next
    html = html & "</table>"
    ecmTeaser = html
end Function
```

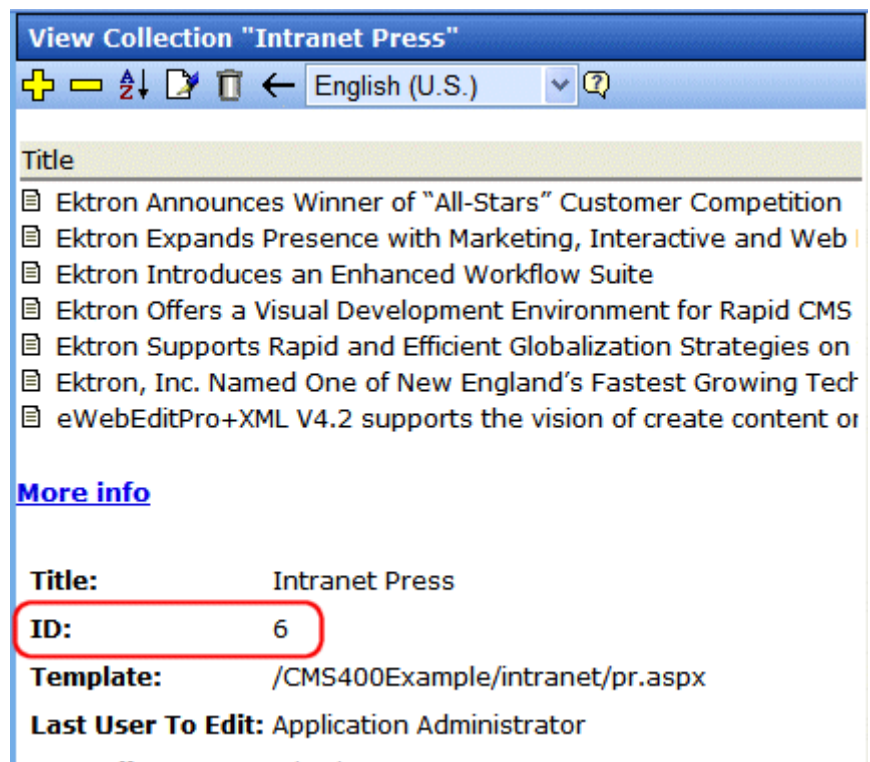
As you can see, the `ecmTeaser` function is an ASP function that creates a table of links to the specified collection. The `ecmTeaser` function displays the:

- Content title
- Date modified
- Content block summary

Collection Attributes

ID

The collection ID number is assigned to the collection when it is created.



DisplayFunction

A display function determines which elements of the content blocks in a collection appear on a Web page. Via ASP scripting, you can create display functions or use any of the three included functions. You must define the display function in two places.

- In the ecmCollection function as a case
- In the API file as a function

RSS Collection Function

The RSS collection function is used to create an XML list of links that were created as a collection in the Ektron CMS400.NET Workarea. The `ecmRssCollection` function is highly customizable, allowing you to easily define how the collection data appears on the Web page.

Shown here is the format of the `ecmRssCollection` function.

```
<% ecmRssCollection "id" %>
```

The `ecmRssCollection` function attributes are explained below.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.

See Also: The **Ektron CMS400.NET** User Manual Section "Working with Collections" > "Comparison of Collections, Menus, and the List Summary Features"

Here is how an RSS Collection appears when viewed without an RSS reader on a web page.

```

<?xml version="1.0" ?>
<rss xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" version="2.0">
  <channel>
    <title>Books</title>
    <link>http://localhost/cms400demo/WorkArea/WebServices/ContentWS.asmx</link>
    <description></description>
    <item>
      <title>Visual Basic.NET How to Program</title>
      <link>http://localhost/cms400demo/collection.aspx?id=23</link>
      <description>Visual Basic .NET How to Program, Second Edition is an awesome resource for
the beginning VB.NET developer who is serious about the craft of programming. In 1,500 pages, this book
covers an incredible amount of territory, from the evolution of programming languages and web
development all the way to advanced data structures and web services.</description>
      <author>Application Administrator</author>
      <pubDate>3/3/2005 5:15:30 PM</pubDate>
    </item>
    <item>
      <title>Simply Visual Basic.NET 2003</title>
      <link>http://localhost/cms400demo/collection.aspx?id=26</link>
      <description>An Application-Driven Tutorial Approach</description>
      <author>Application Administrator</author>
      <pubDate>1/8/2005 10:50:30 AM</pubDate>
    </item>
    <item>
      <title>Microsoft ASP.NET Programming with Microsoft Visual C#.NET</title>
      <link>http://localhost/cms400demo/collection.aspx?id=25</link>
      <description>Teach yourself how to write high-performance Web applications with ASP.NET and
Visual C# .NET--one step at a time.</description>
      <author>Application Administrator</author>
      <pubDate>1/4/2005 7:16:21 PM</pubDate>
    </item>
    <item>
      <title>HTML for the World Wide Web with XHTML and CSS</title>
      <link>http://localhost/cms400demo/collection.aspx?id=27</link>
      <description>The easiest HTML book we've seen that still manages to be comprehensive, and to
stay firmly grounded in the reality of today's demanding users and complex browsers.</description>
      <author>Application Administrator</author>
      <pubDate>1/19/2005 4:18:06 PM</pubDate>
    </item>
  </channel>
</rss>

```

XML Collection Function

The XML collection function wraps XML around the parameters of a list of content that's created as a collection in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the collection information. The ecmCollectionXML function allows you to easily define how the collection data appears on the Web page.

Shown here is the format of the ecmCollectionXML function.

```
<% ecmCollectionXML      "id", "GetHtml"      %>
```

The ecmCollectionXML function attributes are explained below.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
GetHtml	<p>Determines whether the HTML contained within the content is included in the XML output.</p> <p>0 = Do not get HTML 1 = Get HTML</p> <hr/> <p><i>Note: Ektron recommends using 0 (zero). HTML content can be large and this could cause display problems when your XML collection is displayed on your Web page.</i></p> <hr/>

See Also: The **Ektron CMS400.NET** User Manual Section "Working with Collections" > "Comparison of Collections, Menus, and the List Summary Features"

When using the XML Collection function, the following tags are created around the properties of each item in the collection.

XML Tags	Description
<Content></Content>	Declares a new section of content in the XML collection.
<ID></ID>	<p>The content block ID.</p> <p>For example:</p> <pre><ID>27</ID></pre>

XML Tags	Description
<code><Title></Title></code>	The title of the content block. For example: <code><Title>HTML for the World Wide Web with XHTML and CSS</Title></code>
<code><QuickLink></QuickLink></code>	The quicklink associated with the content block. For example: <code><QuickLink>/cms400demo/collection.aspx?id=27</QuickLink></code>
<code><Teaser></Teaser></code>	The summary of the content block. For example: <code><Teaser>The easiest HTML book weve seen that still manages to be comprehensive.</Teaser></code>
<code><Html></Html></code>	All of the HTML content contained in a content block. See Also: "GetHtml" on page 471 For example: <code><Html>ASP.NET Unleashed, Second Edition is really big, really thorough, and really <i>good</i>. The explanations are simple and clear as anything weve seen in a .NET book. The sample code addresses virtually every facet of ASP.NET development, from little gems (code for validating credit cards) to full-fledged web stores. Best of all, Stephen Walther identifies the mistakes ASP.NET developers are most likely to make -- and presents detailed solutions. </Html></code>
<code><StartDate></StartDate></code>	The content block's start date formatted as a .NET date type For example: <code><StartDate>1/1/0001 12:00:00 AM</StartDate></code>
<code><DateModified></DateModified></code>	The last date the content block was modified. For example: <code><DateModified>1/19/2005 4:18:06 PM</DateModified></code>
<code><EndDate></EndDate></code>	The date the content block stops running on the Web site For example: <code><EndDate>12/31/9999 12:00:00 AM</EndDate></code>

XML Tags	Description
<LastEditorFname></LastEditorFname>	The first name of the last person who edited the content block. For example: <LastEditorFname>John</LastEditorFname>
<LastEditorLname></LastEditorLname>	The last name of the last person who edited the content block. For example: <LastEditorLname>Johnson</LastEditorLname>
<Hyperlink></Hyperlink>	The hyperlink shown in the collection for linking to the content block. For example: <Hyperlink>HTML for the World Wide Web with XHTML and CSS</Hyperlink>
<DisplayStartDate></DisplayStartDate>	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.
<FolderID></FolderID>	The ID of the folder in which the content block is contained. For example: <FolderID>0</FolderID>
<ContentStatus></ContentStatus>	The status of the content block.
<Language></Language>	The language of the content block.
<DisplayDateModified></DisplayDateModified>	edited. It is formatted as a string that represents Ektron CMS400's display of the date. For example: <DisplayDateModified>1/19/2005 4:18:06 PM</DisplayDateModified>
<DisplayEndDate></DisplayEndDate>	The content block's end date. It is formatted as a string that represents Ektron CMS400's display of the date. For example: <DisplayEndDate>12/31/9999 12:00:00 AM</DisplayEndDate>

XML Tags	Description
<code><EndDateAction></EndDateAction></code>	What should happen to the content block on its end date. <ul style="list-style-type: none">• archive display• archive expire• refresh report For example: <code><EndDateAction>Archive_Expire</EndDateAction></code>
<code><Comment></Comment></code>	Comments contained in the content block. For example: <code><Comment>move picture below text</Comment></code>

Here is how the raw XML Collection output appears before an XSLT is applied.

```

<Content>
  <ID>24</ID>
  <Title>ASP.NET Unleashed</Title>
  <QuickLink>/cms400demo/collection.aspx?id=24</QuickLink>
  <Teaser><p><em>ASP.NET Unleashed, Second Edition</em> is really big, really thorough, and really <i>good</i>.</p></Teaser>
  <Html></Html>
  <StartDate>1/1/0001 12:00:00 AM</StartDate>
  <DateModified>1/12/2005 7:53:31 PM</DateModified>
  <EndDate>12/31/9999 12:00:00 AM</EndDate>
  <LastEditorFname>Application</LastEditorFname>
  <LastEditorLname>Administrator</LastEditorLname>
  <Hyperlink><a href="/cms400demo/collection.aspx?id=24">ASP.NET Unleashed</a></Hyperlink>
  <DisplayStartDate></DisplayStartDate>
  <FolderID>0</FolderID>
  <ContentStatus></ContentStatus>
  <Language>0</Language>
  <DisplayDateModified>1/12/2005 7:53:31 PM</DisplayDateModified>
  <DisplayEndDate></DisplayEndDate>
  <EndDateAction>Archive_Expire</EndDateAction>
  <Comment></Comment>
</Content>
<Content>
  <ID>25</ID>
  <Title>Microsoft ASP.NET Programming with Microsoft Visual C#.NET</Title>
  <QuickLink>/cms400demo/collection.aspx?id=25</QuickLink>
  <Teaser><p>Teach yourself how to write high-performance Web applications with ASP.NET and Visual C# .NET--one step at a time.</p></Teaser>
  <Html></Html>
  <StartDate>1/1/0001 12:00:00 AM</StartDate>
  <DateModified>1/4/2005 7:16:21 PM</DateModified>
  <EndDate>12/31/9999 12:00:00 AM</EndDate>
  <LastEditorFname>Application</LastEditorFname>
  <LastEditorLname>Administrator</LastEditorLname>
  <Hyperlink><a href="/cms400demo/collection.aspx?id=25">Microsoft ASP.NET Programming with Microsoft Visual C#.NET</a></Hyperlink>
  <DisplayStartDate></DisplayStartDate>
  <FolderID>0</FolderID>
  <ContentStatus></ContentStatus>
  <Language>0</Language>
  <DisplayDateModified>1/4/2005 7:16:21 PM</DisplayDateModified>
  <DisplayEndDate></DisplayEndDate>
  <EndDateAction>Archive_Expire</EndDateAction>
  <Comment></Comment>
</Content>

```

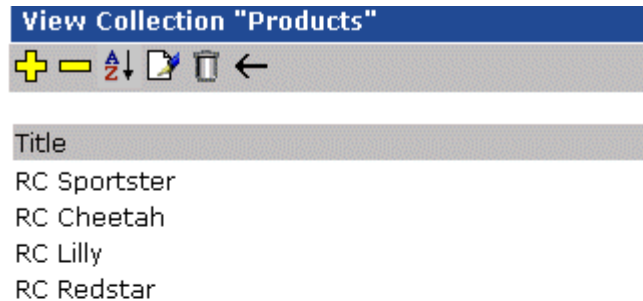
Random Content

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

```
<% ShowRandomContent "id", XSLTid %>
```

Where the "id" represents the ID number of the collection the content blocks, that will be displayed, belong to.



[More info](#)

Title:	Products
ID:	1
Template:	/CMS9999/sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM

And "XSLTid" is the ID number of the display XSLT that will be applied to the content blocks in the collections.

Random Summary

Similar to the random content function, the random summary function randomly displays a content block summary that belongs in a specified collection.

Here is the format for the random content function

```
<% ShowRandomSummary "id" %>
```

The "id" represents ID number of the collection to which the content blocks belong.

View Collection "Products"

Title

RC Sportster

RC Cheetah

RC Lilly

RC Redstar

[More info](#)

Title	Products
ID:	1
Template:	/CMS0000/sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
Product ID:	1

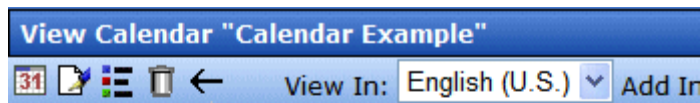
Event Calendar Function

The ecmEvtCalendar function displays a month view event calendar that has been created in Ektron CMS400.NET. Here is the format for the event calendar custom function.

```
<% ecmEvtCalendar "id" %>
```

NOTE You can use ecmCalendar to call a calendar and display a different view, such as daily. For more information, see "Calendar Function" on page 481

The "id" argument represents the ID number of the event calendar in the Workarea.



Title: Calendar Example

ID 1

Description

Location Location:

Start Time Start:

End Time End:

Table Properties:

Event Calendar Style Sheet



































































Before you insert an event Calendar into your Ektron CMS400.NET template, you should make a reference to the style sheet, \CMS400Demo\Workarea\csslib\calendarStyles.css, in your template. This style sheet contains styles that define the Calendar.

Event Calendar Sample

Here is an example of an event Calendar in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <% ecmEvtCalendar "1" %>
  </td>
</tr>
```

When this code is displayed in a browser, the Calendar with an ID-1 is displayed.
Here is the calendar displayed on a Web Page

If...	Calendar Display																																																							
Logged in	<div><div><< May</div><div>June, 2005</div><div>July</div><table><tr><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th></tr><tr><td></td><td></td><td> 1</td><td> 2</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td> 6</td><td> 7</td><td> 8</td><td> 9</td><td></td></tr><tr><td></td><td></td><td>Centaur's Online Marketing 2005</td><td>Centaur's Online Marketing 2005</td><td></td></tr><tr><td> 13</td><td> 14</td><td> 15</td><td> 16</td><td></td></tr><tr><td></td><td>Internet World</td><td></td><td></td><td></td></tr><tr><td> 20</td><td> 21</td><td> 22</td><td> 23</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td> 27</td><td> 28</td><td> 29</td><td> 30</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table></div>	Monday	Tuesday	Wednesday	Thursday	Friday			 1	 2							 6	 7	 8	 9				Centaur's Online Marketing 2005	Centaur's Online Marketing 2005		 13	 14	 15	 16			Internet World				 20	 21	 22	 23							 27	 28	 29	 30						
Monday	Tuesday	Wednesday	Thursday	Friday																																																				
		 1	 2																																																					
 6	 7	 8	 9																																																					
		Centaur's Online Marketing 2005	Centaur's Online Marketing 2005																																																					
 13	 14	 15	 16																																																					
	Internet World																																																							
 20	 21	 22	 23																																																					
 27	 28	 29	 30																																																					

If...	Calendar Display																																																																																		
Logged Out	<table><tr><td colspan="5"><div>< May</div><div>June, 2005</div><div>July >></div></td></tr><tr><td colspan="2">Monday</td><td colspan="2">Tuesday</td><td>Wednesday</td><td>Thursday</td><td>Friday</td></tr><tr><td colspan="2"></td><td colspan="2"></td><td>1</td><td>2</td><td>3</td></tr><tr><td colspan="2"></td><td colspan="2"></td><td></td><td></td><td></td></tr><tr><td>6</td><td></td><td>7</td><td></td><td>8</td><td>9</td><td>10</td></tr><tr><td colspan="2"></td><td colspan="2"></td><td>Centaur's Online Marketing 2005</td><td>Centaur's Online Marketing 2005</td><td></td></tr><tr><td>13</td><td></td><td>14</td><td></td><td>15</td><td>16</td><td>17</td></tr><tr><td colspan="2"></td><td colspan="2">Internet World</td><td></td><td></td><td></td></tr><tr><td>20</td><td></td><td>21</td><td></td><td>22</td><td>23</td><td>24</td></tr><tr><td colspan="2"></td><td colspan="2"></td><td></td><td></td><td></td></tr><tr><td>27</td><td></td><td>28</td><td></td><td>29</td><td>30</td><td></td></tr><tr><td colspan="2"></td><td colspan="2"></td><td></td><td></td><td></td></tr></table>	<div>< May</div> <div>June, 2005</div> <div>July >></div>					Monday		Tuesday		Wednesday	Thursday	Friday					1	2	3								6		7		8	9	10					Centaur's Online Marketing 2005	Centaur's Online Marketing 2005		13		14		15	16	17			Internet World					20		21		22	23	24								27		28		29	30								
<div>< May</div> <div>June, 2005</div> <div>July >></div>																																																																																			
Monday		Tuesday		Wednesday	Thursday	Friday																																																																													
				1	2	3																																																																													
6		7		8	9	10																																																																													
				Centaur's Online Marketing 2005	Centaur's Online Marketing 2005																																																																														
13		14		15	16	17																																																																													
		Internet World																																																																																	
20		21		22	23	24																																																																													
27		28		29	30																																																																														

As you can see, when you are logged into the Web site, you can add and view Calendar events. When logged out, you can only view calendar events.

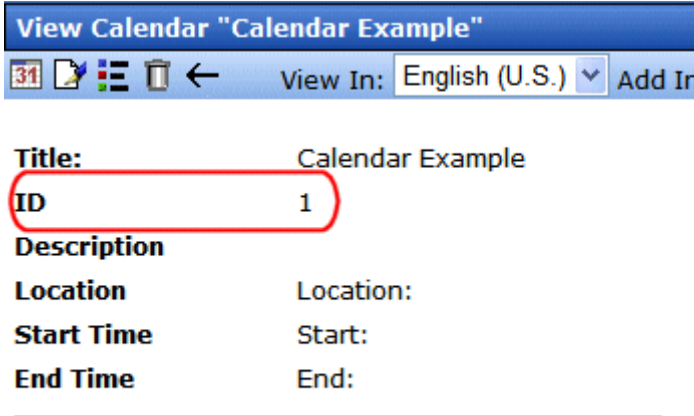
Calendar Function


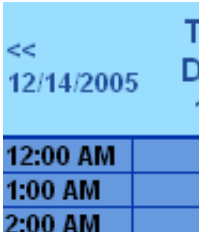


The ecmCalendar function displays a calendar that has been created in Ektron CMS400.NET. Here is the format for the event calendar custom function.

```
<% ecmCalendar "id", "DisplayType" %>
```

NOTE While you can display a month view calendar with ecmCalendar, you can also use ecmEvtCalendar to display a month view calendar. For more information, see "Event Calendar Function" on page 478

The ecmCalendar function attributes are explained below.

Attribute	Description
id	<p>The "id" argument represents the ID number of the event calendar in the Workarea.</p>  <p>Title: Calendar Example</p> <p>ID 1</p> <p>Description</p> <p>Location Location:</p> <p>Start Time Start:</p> <p>End Time End:</p> <p>Table Properties:</p>

Attribute	Description
DisplayType	<p>Specify the Calendar display type.</p> <ul style="list-style-type: none"> Month - A 30 day calendar month view.  Day - Single day view.  Inline - Display events that occur from the start date of the calendar through the end date.  MonthUpWithEvent - Displays a month calendar. Events appear to the right of the calendar when you click on a day. 

Calendar Style Sheet

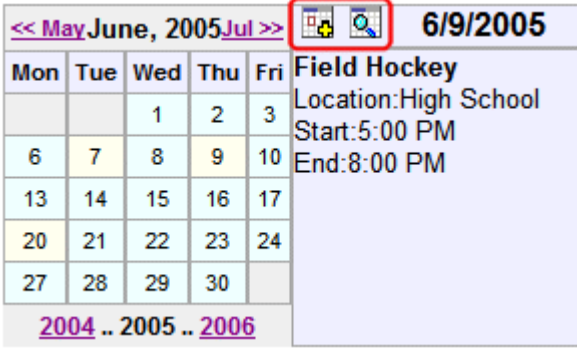
Before you insert an event Calendar into your Ektron CMS400.NET template, you should make a reference to the style sheet, `\CMS400Demo\Workarea\csslib\calendarStyles.css`, in your template. This style sheet contains styles that define the Calendar.

Event Calendar Sample

Here is an example of an event Calendar in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <% ecmCalendar "1", "MonthUpWithEvent" %>
  </td>
</tr>
```

When this code is displayed in a browser, the Calendar with an ID of 1 and DisplayType of MonthUpWithEvent is displayed. Here is the calendar displayed on a Web Page

If...	Calendar Display
Logged in	<p>Notice the add event and view event buttons that appear when logged in.</p> 

If...	Calendar Display																																																
Logged Out	<table><tr><td colspan="5"><< May June, 2005 Jul >></td><td>6/9/2005</td></tr><tr><td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</td><td>Fri</td><td>Field Hockey</td></tr><tr><td></td><td></td><td>1</td><td>2</td><td>3</td><td>Location:High School</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>Start:5:00 PM</td></tr><tr><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>End:8:00 PM</td></tr><tr><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td></td></tr><tr><td>27</td><td>28</td><td>29</td><td>30</td><td></td><td></td></tr><tr><td colspan="5">2004 .. 2005 .. 2006</td><td></td></tr></table>	<< May June, 2005 Jul >>					6/9/2005	Mon	Tue	Wed	Thu	Fri	Field Hockey			1	2	3	Location:High School	6	7	8	9	10	Start:5:00 PM	13	14	15	16	17	End:8:00 PM	20	21	22	23	24		27	28	29	30			2004 .. 2005 .. 2006					
<< May June, 2005 Jul >>					6/9/2005																																												
Mon	Tue	Wed	Thu	Fri	Field Hockey																																												
		1	2	3	Location:High School																																												
6	7	8	9	10	Start:5:00 PM																																												
13	14	15	16	17	End:8:00 PM																																												
20	21	22	23	24																																													
27	28	29	30																																														
2004 .. 2005 .. 2006																																																	

As you can see, when you are logged into the Web site, you can add and view Calendar events. When logged out, you can only view calendar events.

Form Function

The `ecmFormBlock` and `ecmFormBlockNoFormTag` functions display an HTML form content block. Both functions have the same parameters and display the form the same way. The only difference is `ecmFormBlock` automatically inserts form tags, `ecmFormBlockNoFormTag` does not.

The following is an example of form tags.

```
<form name="myform" action="#" onsubmit="return  
EkFmValidate(this);" method="post">
```

```
</form>
```

When using `ecmFormBlock`, form tags are added automatically. When using `ecmFormBlockNoFormTag`, form tags need to be added manually.

For more information about HTML forms, see the **Ektron CMS400.NET** Administrator manual chapter “Working with HTML Forms.”

IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the Ektron CMS400.NET Administrator manual chapter “Managing Library Assets”, section “Updating Default Template for Multiple Quicklinks.”

Here are the two formats for the form custom function.

```
<% ecmFormBlock “id” %>  
or  
<% ecmFormBlockNoFormTag “id” %>
```

NOTE

As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see “MultiPurpose Function” on page 487.

The "id" argument represents the form's ID number, visible when viewing the form from the View Form screen.

ecmFormBlock and ecmFormBlockNoFormTag are interchangeable when using this function. Here is an example of a form content block in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <% ecmFormBlock "367" %>
  </td>
</tr>
```

When this code is read by a browser, the following is displayed.

Vacation Request Form

***Required Fields**

When a visitor to your site submits a form, the form uses the form information to process the data.

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or an content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function:

`<% ecmMultiPurpose "id" %>`

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

`http://localhost/cms400sample/index.asp?id=2`

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

`http://localhost/cms400sample/index.asp?ekfrm=2`

NOTE For more information about passing the id value as a url parameter, see ["Dynamic Content Block" on page 419](#).

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls ecmContentBlock and passes the DefaultID value as the id argument. If the DefaultID value is zero (0), ecmContentBlock is not called.

Metadata Function

The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS400.NET Web site. This custom tag lets you generate the list of metadata edited by the authors and insert it in the source of the Web page. Below is an example of the metadata custom tag.

```
<% ecmMetaData "ItemList","Spare1" %>
```

The attributes are described below.

Attribute	Description	Required?
ItemList	List of the content block ID numbers, and exclusions See Also: "Creating an Item List" on page 488 and "Item List" on page 491	No
Spare1	Not yet implemented See Also: "Spare 1" on page 493	MUST BE ""

Creating an Item List

The Item List for the ecmMetadata tag has the following format.

```
"[content block id; Exclusion list]...[content block id; Exclusion list]"
```

Attribute	Description	Required?
[(left bracket)	Delineates the beginning of a content block id/exclusion list pair	Yes
Content Block ID	The ID of the content block whose metadata you want to retrieve	Yes
; (semicolon)	Separator: Separates content block id from the exclusion list	No (Yes if including an exclusion list)
Exclusion List	A comma delimited list of metadata you do not want to be retrieved for this content block.	No
] (right bracket)	Delineates the end of a content block id/exclusion list pair	Yes

ecmMetadata Example

For this example, our administrator has defined the following metadata types for Ektron CMS400.NET.

Name	Style	Type	Editable	Required	Remove Dups	Case Sensitive
Title	n/a	HTML	Yes	Yes	No	n/a
Keywords	Name	Meta	Yes	Yes	Yes	No
Content-Type	http-equiv	Meta	No	Yes	Yes	n/a

Our example Web page has two content blocks displayed.

- content block 1, the company's home page introduction text
- content block 2, a sidebar containing the lead story for the week

The metadata entered by your content editors for these two content blocks are:

Content Block 1:

Title – "Welcome to Our Home Page"

Keywords – "Home; page; company; Welcome"

Content-Type -- "text/html; charset=iso-8859-1"

Content Block 2:

Title – "Announcing our New Product!"

Keywords – "Announce; company; product; widget"

Content-Type -- "text/html; charset=iso-8859-1"

To include both content blocks' metadata in a Web page, the ecmMetadata call looks like this.

```
<html>
  <head>
    <% ecmMetadata "[1][2;Title,Content-Type]", "" %>
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

In the above example, the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). When the Web page is viewed, the Web page source contains the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
    product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```

    (rest of head)
</head>
<body>
    (rest of web page document)
</body>
</html>

```

The ecmMetadata constructed three HTML tags based on the administrator defined parameters and the ecmMetadata call.

The metadata tag was constructed in the page because:

- **[1]** - Displays all metadata content for the content block ID=1. In this example, the title, keywords, and content-type were defined, so they were displayed as:

```

<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
    Announce; product; widget">
<meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">

```

- **[2;title,content-type]** - Displays all metadata content for content block ID=2, except for its title and content-type definitions.

```

<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
    Announce; product; widget">
<meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">

```

The next section provides more examples.

More ecmMetadata Examples

Shown below is an example of the metadata function in an ASP Web page.

```

<TR>
    <TD>
        <% ecmMetaData "[1][2;title][14;keywords]", "" %>
    </TD>
</TR>

```

The previous example has the following properties:

- All metadata for content block ID=1 is inserted into the page
- All metadata for the content block ID=2 except for its title is inserted into the page
- All metadata for the content block ID=14 except for its keywords is inserted into the page

Using a Dynamic Metadata Tag

You can use a metadata tag in a template to define the metadata for a specific content block. However, you can also use a dynamic metadata tag combined with a content block tag to show the metadata for all content blocks on a Web page.

This allows you to create a template and not have to keep updating the tag by adding the new ID and exclusion list. Shown below is the sample code for this.

```
<!-- #include virtual="/CMS400.NET/sample/CMS400.NET/scripts/ektronAPI.asp" -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <% Dim id
      if (request.QueryString("id") <> "") then
        id = request.QueryString("id")
      else
        id = 1
      end if
      ecmMetadata "[" & id & "]", ""
    %>
  </head>
  <body>
    <% ecmContentBlock(request.QueryString("id")) %>
  </body>
</html>
```

In this example, if you specify a content block in the URL and the page is displayed, the metadata takes that ID and displays the proper metadata in the source for the content block.

ecmMetadata Attributes

Below are the attributes and their descriptions.

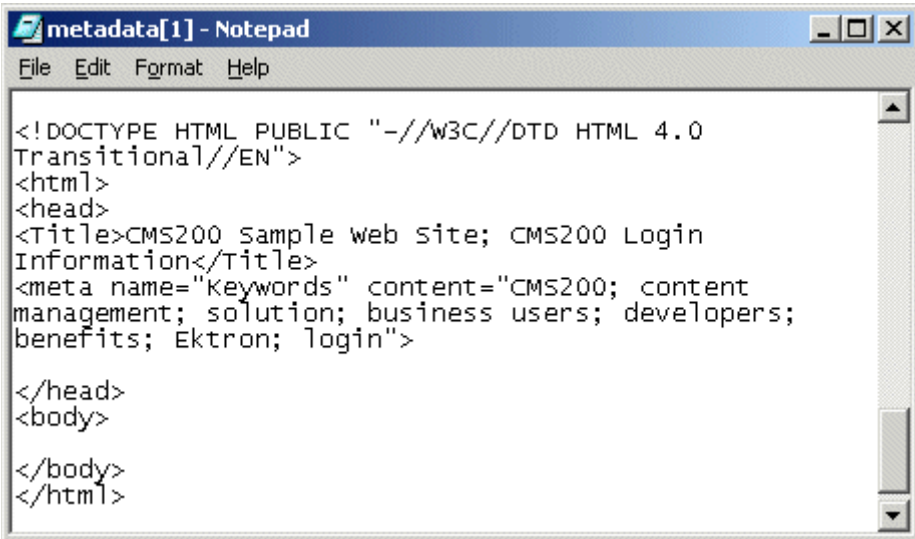
Item List

In this attribute of the ecmMetadata tag, you need to specify two values.

Value	Specifies
ID	The content block ID whose metadata will appear
Exclusion List	The metadata to be excluded. See Also: "Creating an Item List" on page 488

Some examples might be:

```
<% ecmMetaData "[1][2]", "" %>
```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site; CMS200 Login
Information</Title>
<meta name="keywords" content="CMS200; content
management; solution; business users; developers;
benefits; Ektron; login">

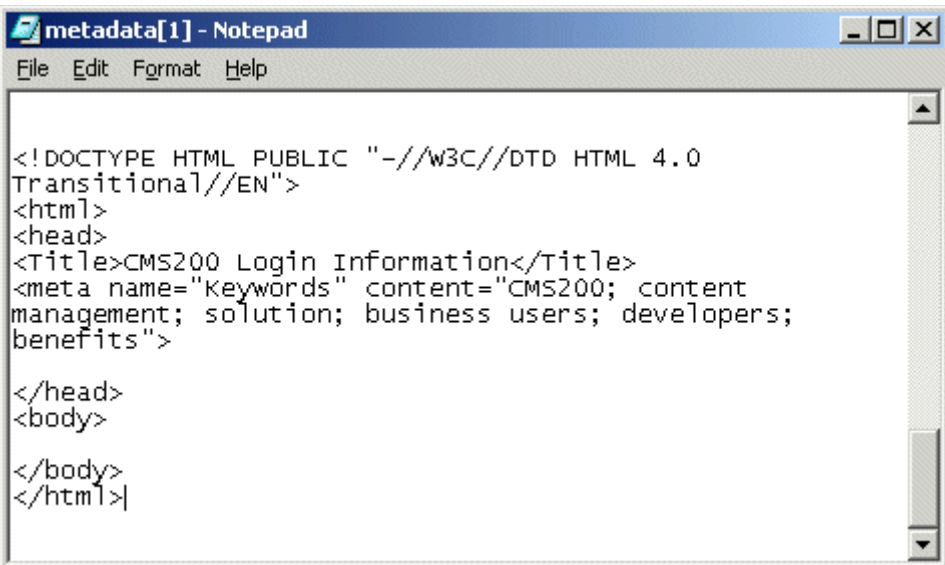
</head>
<body>

</body>
</html>

```

In this example, all metadata from content block ID=1 and 2 is displayed.

```
<% ecmMetaData "[1;title][2;keywords]", "" %>
```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

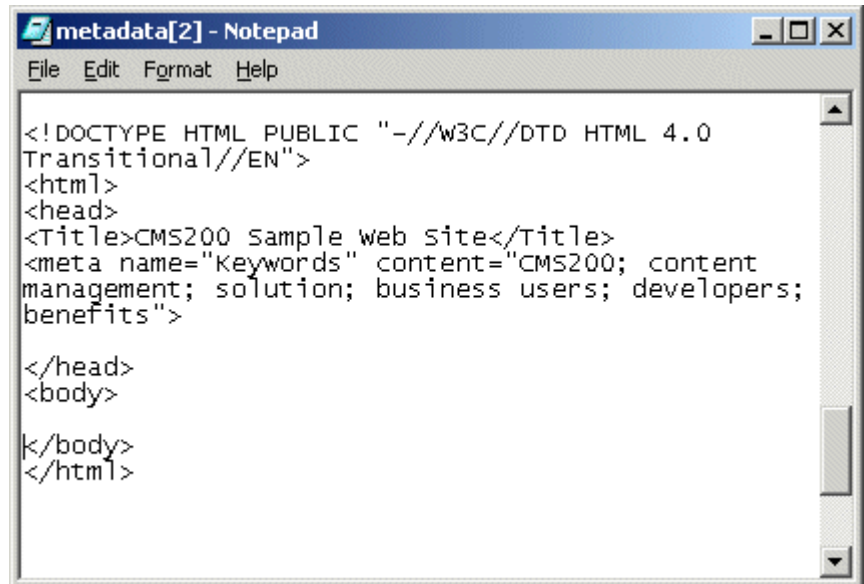
</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed *except* for the title. Also, all the metadata for content block ID=2 is displayed *except* its keywords.

```
<% ecmMetaData "[1][2:title,keywords]","", "" %>
```

A screenshot of a Notepad window titled "metadata[2] - Notepad". The window has a menu bar with "File", "Edit", "Format", and "Help". The text area contains the following HTML code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">  
<html>  
<head>  
<title>CMS200 Sample web site</Title>  
<meta name="keywords" content="CMS200; content  
management; solution; business users; developers;  
benefits">  
  
</head>  
<body>  
  
</body>  
</html>
```

In this example, all metadata from content block ID=1 is displayed, and all metadata from content block ID=2 is displayed *except* its title and keywords.

Spare 1

One attribute field at the end is currently not used. It is for future scalability. When you insert a single metadata tag into a template, you must leave this attributes as "".

Show Site Language Function

This function lets a visitor to your Web site select a language in which to view the site. Here is what the function looks like when published on a Web page.



It lists all languages selected on the **Settings > Configuration > Language settings** screen. (For more information, see the “Multi-Language Support” chapter of the **Ektron CMS400.NET** Administrator manual.)

Below is an example of the Show Site Language custom tag.

```
<% ecmShowSiteLanguage(Fieldname)%>
```

You can place this function in any location of any page on your site. By default, it appears in the lower right corner of the home page.

NOTE [This function only works with Asp and ASP.Net.](#)

The attribute is described in this table.

Attribute	Description
Fieldname	<p>An optional argument. You can leave it blank and allow the drop-down list to be populated by the available languages for your Web site. For example:</p> <pre><%=ecmShowSiteLanguage('')%></pre> <p>Or you can pass in an argument such as:</p> <pre><%=ecmShowSiteLanguage('mysitelanguage')%></pre>

DHTML Menus

Ektron CMS400.NET's Menu feature lets users create and maintain a dropdown menu system for your Web site. The Menu feature lets you display menu options that link to content blocks, library assets, external hyperlinks and submenus.

To learn more about menus, please refer to the "Menu Feature" chapter of the **Ektron CMS400.NET** User Manual.

Additional Documentation in Ektron Knowledge Base

This chapter explains how to implement standard menus that ship with **Ektron CMS400.NET**. Ektron's Web site also provides Knowledge Base articles that explain how to extend the standard menu features.

Displaying a Menu on a Web Page

GenerateMenu and DropMenu

IMPORTANT!

The menu functions described below are only one example of how to implement menus. The scripts reside in the Application API file. You can implement your own menu system using JavaScript.

Use the `GenerateNextGenMenu` and `DropNextGenMenu` custom functions to display a DHTML menu on a Web page. Place `GenerateNextGenMenu` once on the page to provide an infrastructure for `DropNextGenMenu`. Then, place `DropNextGenMenu` as needed to implement individual menus.

Follow these steps to display a menu on a Web template.

1. Place the following code directly after the page's `<body>` tag.

```
<%= (ecmDHTML_GenerateNextGenMenu( "menu id number" )) %>
```

For example:

```
<%= (ecmDHTML_GenerateNextGenMenu( "1" )) %>
```

To display more than one menu, separate each id number with a comma. For example:

```
<%= (ecmDHTML_GenerateNextGenMenu( "1, 2, 3" )) %>
```

2. If the user clicks *text* to display the menu, place the following code within the template file where the text would appear.

```
<%= ecmDHTML_DropNextGenMenu(menu id number,"text user clicks to display menu")%>
```

For example:

```
<%= ecmDHTML_DropNextGenMenu(1,"Products")%>
```

If the user clicks a *graphic* to invoke the menu, use this syntax:

```
<%= ecmDHTML_DropNextGenMenu(menu id number,'<img src=graphic file name>')%>
```

For example:

```
<%= ecmDHTML_DropNextGenMenu(1,'<img src=mymenugraphic.jpg>')%>
```

Displaying Menus in a Multi-language System

If your Web site supports multiple languages, and the user selects a language before viewing the menu, if an edition of the menu is available in that language, it appears. If not, nothing appears.

For more information, see the **Ektron CMS400.NET** Administrator Manual section “Multi-Language Support” > “Working with Multi-Language Content” > “Working with Menus in a Multi-Language System.”

Customizing the Menu's Appearance

You can customize a menu's appearance, such as the font style, color, and size. So, you have the flexibility to coordinate a menu with the design your Web site.

To customize a menu in **Ektron CMS400.NET**, you can change the `ek_menu.js` file. See Also: ["ek_Menu.js" on page 496](#)


ek_Menu.js

You can change the look and feel of a menu by changing the `ek_menu.js` file. If you used the default settings when installing **Ektron CMS400.NET**, this file is in `webroot/cms400demo/Workarea/java/`.

Use the following table to help you change menu attributes.

Attribute name	Specifies the	Default value
fontSize	Size of menu text	14
fontWeight	Thickness of menu text You can set FontWeight by name (Bold or Plain) or value. Values range from 0 -1000.	Plain

Attribute name	Specifies the	Default value
fontFamily	Font style of menu text <i>See Also: "FontFamily" on page 498</i>	arial, helvetica, espy, sans-serif
fontColor	Color of menu text <u>Note: When changing the color, you can specify the color name or code.</u>	#000000
fontColorHiLite	Color of text when menu item is highlighted	#ffffff
bgColor	Color of border between menu items	#555555
menuBorder	Width (in pixels) of border around main menu	1
menuItemBorder	Width (in pixels) of border around each menu item	1
menuItemBgColor	Background color of menu items	#6699ff
menuLiteBgColor	Color of main menu border highlight <i>See Also: "menuLiteBgColor" on page 498</i>	#ffffff
menuBorderBgColor	Color of the main menu border	#6699ff
menuHiLiteBgColor	Background color of highlighted menu item	#000084
menuContainerBgColor	Background color of the <i>container</i> <u>Note: The container is like a <div> tag that displays the menu. When you set the container color, it appears at the top and left side of the menu frame only.</u>	#6699ff

Attribute name	Specifies the	Default value
childMenuIcon	<p>Path to the submenu icon (the image used to indicate that a submenu is available from this menu item).</p>  <p>To display a different image, either provide an external path to it or upload the image to Ektron CMS400.NET then provide a local path.</p>	./Workarea/ images/ application/ arrows.gif
childMenuIconHiLite	<p>The path to the icon that appears when you hover the mouse over a menu item that leads to a submenu.</p> <p>To display a different image, either provide an external path to it or upload the image to Ektron CMS400.NET then provide a local path.</p>	./Workarea/ images/ application/ arrows2.gif

FontFamily

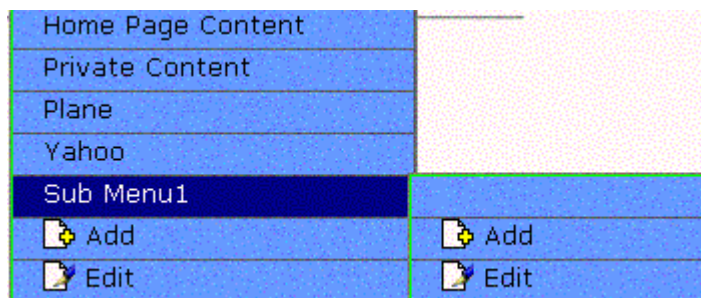
Use the fontFamily attribute to specify the menu font. It must be supported by the browser.

You can assign to this property several, comma-separated values.

NOTE To display a font family other than the browser default, specify only the font name and no other comma-separated values. If you specify more than one value for the font family, Ektron CMS400.NET uses the default IE settings.

menuLiteBgColor

Example



XML Menu

The ecmGetMenuXML function wraps XML around the properties of a menu created in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the menu. The ecmGetMenuXML function allows you to easily define how the menu data appears on the Web page.

Shown here is the format of the ecmGetMenuXML function.

```
<% ecmGetMenuXML (id) %>
```

The ecmGetMenuXml function attributes are explained below.

Attribute	Description
Id	The id that associated with the menu.

Here is how the raw XML Menu output appears before an XSLT is applied.

```
<Item>
  <ItemID>1</ItemID>
  <ItemType>Submenu</ItemType>
  <ItemTitle>MenuExample</ItemTitle>
  <ItemImage></ItemImage>
  <ItemImageOverride>False</ItemImageOverride>
  <Menu>
    <ID>1</ID>
    <Title>MenuExample</Title>
    <Template></Template>
    <Type>content</Type>
    <Link></Link>
    <ParentID>0</ParentID>
    <AncestorID>1</AncestorID>
    <FolderID>0</FolderID>
    <Description></Description>
    <ImageOverride>False</ImageOverride>
  </Item>
  <Item>
    <ItemID>2</ItemID>
    <ItemType>Submenu</ItemType>
    <ItemTitle>Book</ItemTitle>
    <ItemImage></ItemImage>
    <ItemImageOverride>False</ItemImageOverride>
    <Menu>
      <ID>2</ID>
      <Title>Book</Title>
      <Template></Template>
      <Type>content</Type>
      <ParentID>1</ParentID>
      <ImageOverride>False</ImageOverride>
    <Item>
      <ItemID>23</ItemID>
      <ItemType>content</ItemType>
      <ItemTitle>VB.NET How to</ItemTitle>
      <ItemImage></ItemImage>
      <ItemLink>/cms400demo/collection.aspx?id=23</ItemLink>
      <ItemTarget>2</ItemTarget>
    </Item>
  </Menu>
</Item>
```

Custom ColdFusion Functions

Ektron CMS400.NET maintains and display content through the use of custom Ektron ColdFusion functions and ComObjects.

When using custom ColdFusion functions, the following paths must be updated in the application.cfm.

```
<cfset application.WSDLpath = "http://localhost/cms400demo/WorkArea/WebServices/  
ContentWS.asmx?WSDL" />
```

and

```
<cfset application.AppPath = "/cms400demo/WorkArea/platforms/ColdFusion/" />
```

The `application.WSDLpath` needs to be updated with the location of the WSDL file. For example, if your site is `www.myelektronsite.com`, the `WSDLpath` would be `http://www.myelektronsite.com/WorkArea/WebServices/ContentWS.asmx`.

The `application.AppPath` needs to be updated with the path to the ColdFusion custom Functions.

To customize your Ektron CMS400.NET application, you will need to be familiar with these tags

Function	Description	More Information
Calendar Functions		
Event Calendar	Displays a month view event calendar	"Event Calendar Function" on page 513
Collection Functions		
Collection	Displays links generated by collections	"Collection Function" on page 509
CollectionXML	Creates XML for a collection by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML Collection Function" on page 513
Random Content	Randomly displays the content of a collection item	"Random Content Function" on page 505
Random Summary	Randomly displays the summary of a collection item	"Random Summary Function" on page 506
RSS Collection	Creates an RSS feed out of the items in a collection.	"RSS Collection Function" on page 513

Function	Description	More Information
Content Functions		
Add Content	Allows a user to add content to Ektron CMS400.NET programmatically.	"Add Content" on page 504
Content Block	Displays and manages a content block	"Content Block Function" on page 503
Form	Displays a content block associated with a form	"Form Function" on page 516
Multipurpose	Displays either a content block or a form	"MultiPurpose Function" on page 516
Language Functions		
Language Select	Lets a site visitor select a language in which to view site.	"Language Select Function" on page 520
List Summary Functions		
Archive ListSummary	Displays summaries of archived content blocks	"Archive List Summary Function" on page 509
List Summary	Displays a list of content block summaries	"List Summary Function" on page 507
List Summary XML	Creates XML for a list summary by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML List Summary Function" on page 508
RSS List Summary	Creates an RSS feed out of the items in a list summary.	"RSS List Summary Function" on page 508
Single Summary	Displays the summary for a content block.	"Single Summary Function" on page 509
Login Functions		
Login	Adds the login or logout button	"Login Function" on page 503
Menu Functions		

Function	Description	More Information
DropHTMLMenu	Used to display a DHTML menu	"DHTML Menus" on page 518
GenerateMenu	Works with DropHTML menu to display DHTML menus	"DHTML Menus" on page 518
XML Menu	Creates XML for a menu by wrapping tags around all of the menu items properties.	"XML Menu Function" on page 520
Metadata Functions		
Metadata	Displays the metadata for the current Web page	"Metadata Function" on page 517
Search Functions		
Search	Inserts a text box with a search button	"Search Function" on page 506
Search Display	Creates an area to display search results	"Search Display Function" on page 507

Login Function

This tag is responsible for displaying the following buttons on the Web page.

This function is basically the same as the Login function in ASP. See "[Login Function](#)" on page 414 for details. The only difference is the syntax used to implement the function. This is listed below.

Login Tag Example

```
<tr>
  <td>
    <cfmodule template="#request.ecm.AppPath#ecmLogin.cfm">
  </td>
</tr>
```

Content Block Function

The content block and content blockEX tags are used to display a content block on an Ektron CMS400.NET Web page.

Shown here is the format for the content block function:

```
<CFMODULE template="#request.ecm.AppPath#ecmContentBlock.cfm"
  ID= " "
>
```

Shown here is the format for the content blockEX function:

```
<CFMODULE template="#request.ecm.AppPath#ecmContentBlockEx.cfm"
           ID=""
           XSLT=""
           Override=""
>
```

This functions are basically the same as the ContentBlock and ContentBlockEX function in ASP. See ["Content Block Function" on page 416](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion examples are below.

Static Content Block

```
<tr>
  <td>
    <CFMODULE id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm">
  </td>
</tr>
```

Dynamic Content Block

```
<tr>
  <td>
    <CFMODULE id="#id#" template="#request.ecm.AppPath#ecmContentBlock.cfm">
  </td>
</tr>
```

XML Content Block

```
<tr>
  <td>
    <CFMODULE id="13" XSLT="" override="1"
      template="#request.ecm.AppPath#ecmContentBlockEx.cfm">
  </td>
</tr>
```

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS400.NET without the need of logging in.

Here is the format for the add content function:

```
<CFModule template="#request.ecm.AppPath#ecmAddContent.cfm" Username="",
Password="",
Domain="",
content_title="",
content_comment="",
ContentHtml="",
SummaryHTML="",
ContentLanguage="",
FolderID="",
GoLive="",
EndDate="",
MetaInfo="" >
```

This function is basically the same as the Add Content function in ASP. See ["Add Content" on page 424](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Add Content Example

The following is an example of the add content function in a template:

```
<tr>
  <td>
    <CFModule template="#request.ecm.AppPath#ecmAddContent.cfm "AddContent
    UserName="admin", Password="admin", Domain="", content_title="Content Title",
    content_comment="Comment", ContentHTML="This is the <b>body</b> of the content.",
    SummaryHTML="This is the <i>summary</i> for the content.", ContentLanguage="1033",
    FolderID="0", GoLive="", EndDate="", MetaInfo="">
  </td>
</tr>
```

Random Content Function

The random content function is used to randomly display the content of a content block that belongs to a specified collection. Below is the format for the random content function:

```
<CFMODULE template="#request.ecm.AppPath#ecmShowRandomContent.cfm"
    ID=" "
    XSLT=" "
    Override=" "
>
```

This function is basically the same as the Random Content function in ASP. See ["Random Content" on page 476](#) for details.

Random Summary Function

Similar to the random content function, the random summary function can randomly display a content block summary that belongs to a specified collection. Here is the format for the random content function

```
<CFMODULE template="#request.ecm.AppPath#ecmShowRandomSummary.cfm"
    ID=" "
>
```

This function is basically the same as the Random Summary function in ASP. See ["Random Summary" on page 477](#) for details.

Search Function

The search tag is used to display a text box, with a search button on the template. When seen on the template on the Web page, the users will be allowed to enter text and click on the search button to search through all the content on the Web site.

Shown here is the format for the ecmSearch tag.

```
<CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm"
    StartingFolder=" "
    Recursive=" "
    TargetPage=" "
    TextBoxSize=" "
    MaxCharacters=" "
    ButtonImageSrc=" "
    ButtonText=" "
    FontFace=" "
    FontColor=" "
    FontSize=" "
    Horizontal=" "
>
```

This function is basically the same as the Search function in ASP. See ["Search Functions" on page 428](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Search Tag Example

```
<tr>
    <td>
        <CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm"
            StartingFolder="\marketing" Recursive="1" TargetPage="search-
            display.cfm" TextBoxSize="25" MaxCharacters="200"
            ButtonImageSrc=" " ButtonText="Search" FontFace="Verdana"
            FontColor="##808080" FontSize="2" Horizontal="0">
        </td>
    </tr>
```

Search Display Function

The search display custom tag created the area where all the search results show after the search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<CFMODULE template="#request.ecm.AppPath#ecmSearchDisplay.cfm"
    MaxNumber=" "
    StyleInfo=" "
    ShowDate=" "
>
```

This function is basically the same as the Search Display function in ASP. See ["Search Display Function" on page 438](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Search Display Example

```
<tr>
  <td>
    <CFMODULE template="#request.ecm.AppPath#ecmSearchDisplay.cfm"
      MaxNumber="0" StyleInfo=" " ShowDate="1">
    </td>
</tr>
```

List Summary Function

The list summary function is used to display the list of content summaries on a Web page. Shown here is the format of the list summary function:

```
<CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
    Folder=" "
    MaxNumber=" "
    Recursive=" "
    ShowSummary=" "
    StyleInfo=" "
    OrderBy=" "
    ShowInfo=" "
>
```

This function is basically the same as the List Summary function in ASP. See ["List Summary Function" on page 442](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

List Summary Example

```
<TR>
  <TD>
    <CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
      Folder="\Marketing\News" MaxNumber="0" Recursive="0"
      ShowSummary="1" StyleInfo=" " OrderBy="Title,asc" ShowInfo=" " >
    </TD>
</TR>
```

XML List Summary Function

The XML List Summary function wraps XML around the properties of a list of content, created as a list summary in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the list summary information. The `ecmListSummaryXML` function allows you to easily define how the list summary data appears on the Web page.

```
<TR>
  <TD>
    <CFMODULE template="#request.ecm.AppPath#ecmListSummaryXML.cfm"
      Folder=" " Recursive=" " OrderBy=" "
      MaxNumber=" " ObjType=" " SummaryType=" " >
  </TD>
</TR>
```

This function is basically the same as the List Summary XML function in ASP. See ["XML List Summary Function" on page 450](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

XML List Summary Example

```
<TR>
  <TD>
    <CFMODULE template="#request.ecm.AppPath#ecmListSummaryXML.cfm"
      Folder="\books\" Recursive="0" OrderBy="Title,asc"
      MaxNumber="0" ObjType="folder" SummaryType="alltypes">
  </TD>
</TR>
```

RSS List Summary Function

The RSS summary function is used to create an XML list of content summaries for RSS Syndication. Shown here is the format of the RSS list summary function:

```
<CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
  Folder=" "
  MaxNumber=" "
  Recursive=" "
  OrderBy=" "
>
```

This function is basically the same as the RSS Summary function in ASP. See ["RSS Summary Function" on page 460](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

List Summary Example

```
<TR>
  <TD>
    <CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
      Folder="\books" MaxNumber="0" Recursive="0"
      OrderBy="Title,asc" >
  </TD>
</TR>
```

```
</TD>
</TR>
```

Archive List Summary Function

The Archive list summary function displays on a Web page a list of content summaries for archived content blocks. For example, you might have a current news page and an archived news page. Use the List Summary Function to display current news, and this function to display older news stories.

On the Content Block **schedule** screen, the user must set an **End Date** then choose the Archive and Display option to have the content blocks appear via this function.

This function has the same parameters as the List Summary Function. See ["List Summary Function" on page 507](#) for details.

Single Summary Function

The single summary tag is used to display the summary of one content block in the Ektron CMS400.NET Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm"
    ContentID=" "
    ShowSummary="1"
    StyleInfo=" "
    ShowInfo=" "
>
```

This function is basically the same as the Single Summary function in ASP. See ["Single Summary Function" on page 456](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Single Summary Example

```
<TR>
  <TD>
    <cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm"
      ContentID="14" ShowSummary="1" StyleInfo=" "
      ShowInfo="DateModified">
    </TD>
  </TR>
```

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS400.NET Workarea. The ecmCollection function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the ecmCollection function.

```
<CFMODULE template="#request.ecm.AppPath#ecmCollection.cfm"
```

```

id="2"
displayFunction="ecmTeaser"
>

```

This function is basically the same as the Collection function in ASP. See ["Collection Function" on page 465](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS400.NET sample Web site.

```

<CFMODULE id="1" displayFunction="ecmNavigation"
template="#request.ecm.AppPath#ecmCollection.cfm">

```

As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used. Below is the source code for the display function "ecmNavigation".

```

<cfparam name="Attributes.nId" default="">
<cfparam name="Attributes.href" default="">
<cfparam name="Attributes.target" default="">
<cfparam name="Attributes.startStyle" default="">
<cfparam name="Attributes.endStyle" default="">
<cfset LocalError = "">
<cfobject class="CMS400.NETContentBrv2.CMS400.NETContent" Action="Create" name="cmsObj">
<cfset cNavs =
cmsObj.GetEcmCollectionItems(request.ecm.AppConfStr,Attributes.nId,0,request.ecm.cmsPreview,r
equest.ecm.user_id,request.ecm.site_id,LocalError)>
<cfif cNavs.Count()>
    <cfset nString = "">
    <cfloop collection=#cNavs# Item=nav>
        <cfset nString = nString & "<a href="">
        <cfif len(Attributes.href)>
            <cfif FindNoCase("?", Attributes.hrefs)>
                <cfset nString = nString & Attributes.href & "&id=" &
                nav.Item("ContentID")>
            <cfelse>
                <cfset nString = nString & Attributes.href & "?id=" &
                nav.Item("ContentID")>
            </cfif>
        </cfif>
        <cfset nString = nString & nav.Item("ContentLinks")>
        </cfif>
        <cfset nString = nString & "" " ">
        <cfif len(Attributes.target)>
            <cfset nString = nString & "target="" " & Attributes.target & "" " ">
        </cfif>
        <cfset nString = nString & ">">
        <cfif len(Attributes.startStyle)>
            <cfset nString = nString & Attributes.startStyle>
        </cfif>
        <cfset nString = nString & nav.Item("ContentTitle")>
        <cfif len(Attributes.endStyle)>
            <cfset nString = nString & Attributes.endStyle>

```

```

        </cfif>
        <cfset nString = nString & "</a>">
    </CFLOOP>
    <cfmodule str="#nString#" nId="#Attributes.nId#"
template="#request.ecm.AppPath#ecmshowCollectionsContent.cfm">
</cfif>
<cfset nNavs = "Nothing">
<cfset cmsObj = "Nothing">

```

As you can see, the ecmNavigation function is a simple ASP function that creates a table of the links that belong to the collection specified. The function also displays the title of the content blocks by using the "info("ContentTitle")) building block.

Example Using the ecmTeaser Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS400.NET sample Web site.

```

<CFMODULE id="2" displayFunction="ecmTeaser"
template="#request.ecm.AppPath#ecmCollection.cfm">

```

In this example, the collection with an ID=2 is displayed, and the function "ecmTeaser" defines how to display the collection data on a Web page. Here is how the collection would appear on a Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used. Below is the source code for the display function "ecmTeaser".

```

<cfparam name="Attributes.nID" default="">
<cfparam name="Attributes.href" default="">
<cfparam name="Attributes.ShowSummary" default="">
<cfparam name="Attributes.StyleInfo" default="">
<cfparam name="Attributes.target" default="">
<cfset ErrorString = "">
<cfobject class="CMS400.NETContentBrv2.CMS400.NETContent" Action="Create" name="tObj">

```

```

<cfset cTeasers = tObj.GetEcmCollectionItems(request.ecm.AppConfStr, Attributes.nId, 0,
request.ecm.cmsPreview, request.ecm.user_id, request.ecm.site_id, ErrorString)>
<cfset tString = "<table border=""0"" cellspacing=""0"" cellpadding=""0"" width=""100%"">">
<cfif cTeasers.Count()>
    <cfloop collection=#cTeasers# Item=cT>
        <cfset tString = tString & "<tr><td>">
        <cfset tString = tString & "<a href="">">
        <cfif len(Attributes.href)>
            <cfif FindNoCase("?", Attributes.hrefs)>
                <cfset tString = tString & Attributes.href & "&id=" &
                cT.Item("ContentID")>
            <cfelse>
                <cfset tStringi = tString & Attributes.href & "?id=" &
                cT.Item("ContentID")>
            </cfif>
        <cfelse>
            <cfset tString = tString & cT.Item("ContentLinks")>
        </cfif>
        <cfset tString = tString & "" " ">
        <cfif len(Attributes.target)>
            <cfset tString = tString & "target="" & Attributes.target & "" " ">
        </cfif>
        <cfset tString = tString & ">">
        <cfset tString = tString & cT.Item("ContentTitle") & "</a>" & "&nbsp;(" &
        DateFormat(cT.Item("DateModified"), "mm/dd/yyyy") & " " &
        TimeFormat(cT.Item("DateModified"), "h:mm:ss tt") & ")</td></tr><tr><td>" &
        cT.Item("ContentTeaser")>
        <cfset tString = tString & "</td></tr><tr><td>&nbsp;</td></tr>">
    </cfloop>
    <cfset tString = tString & "</table>">
    <cfmodule str="#tString#" nId="#Attributes.nId#"
template="#request.ecm.AppPath#ecmshowCollectionsContent.cfm">
</cfif>

```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmTeaser function also displays the:

- Content Title
- Date Modified
- Content Teaser

Collection Variables

See ["Collection Attributes" on page 468](#)

DisplayFunction

The display function defines how collection data appears on a Web page. Using ColdFusion scripting, you can create your own display functions, or use the three included functions.

XML Collection Function

The XML collection function wraps XML around the parameters of a list of content that's created as a collection in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the collection information. The `ecmCollectionXML` function allows you to easily define how the collection data appears on the Web page.

Shown here is the format of the `ecmCollection` function.

```
<CFMODULE template="#request.ecm.AppPath#ecmCollectionXML.cfm"
           id=" "
           getHtml=" "
>
```

This function is basically the same as the XML Collection function in ASP. See ["XML Collection Function" on page 471](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

```
<CFMODULE template="#request.ecm.AppPath#ecmCollectionXML.cfm" id="3" getHtml="0">
```

RSS Collection Function

The RSS collection function is used to create an XML list of links that were created as a collection in the Ektron CMS400.NET Workarea and display them in RSS Syndication.

Shown here is the format of the `ecmRssCollection` function.

```
<CFMODULE template="#request.ecm.AppPath#ecmRssCollection.cfm"
           id="2"
>
```

This function is basically the same as the RSS Collection function in ASP. See ["RSS Collection Function" on page 469](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it.

Event Calendar Function

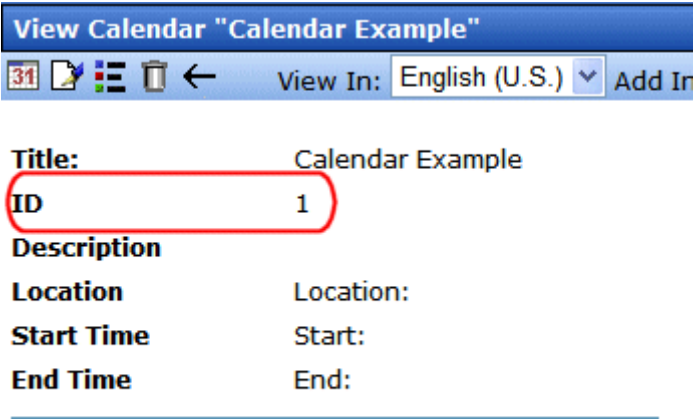
The `ecmEvtCalendar` function is used to display an event calendar that has been created in the Ektron CMS400.NET Workarea.


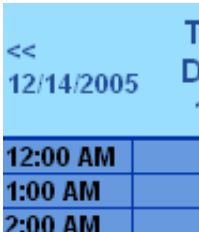


Displayed here is the format for the event calendar custom function:

```
<CFMODULE template="#request.ecm.AppPath#ecmEvtCalendar.cfm" Calendar_ID=" " DisplayType=" ">
```

This function is basically the same as the Event Calendar function in ASP. See ["Event Calendar Function" on page 478](#) for details.

The only differences between the Coldfusion function and the ASP one are the use of the DisplayType parameter and the syntax for implementing it. The ecmEvtCalendar function parameters are defined below.

Attribute	Description
id	<div>The “id” argument represents the ID number of the event calendar in the Workarea.</div> <div><p>The screenshot shows a web interface titled "View Calendar 'Calendar Example'". It includes a navigation bar with a calendar icon, a document icon, a list icon, a trash icon, and a back arrow. The "View In:" dropdown is set to "English (U.S.)". Below the navigation bar, the "Title:" is "Calendar Example". The "ID" field is highlighted with a red circle and contains the value "1". Below the "ID" field, the "Description:" field is empty. Below the "Description:" field, the "Location:" field is empty. Below the "Location:" field, the "Start Time:" field is empty. Below the "Start Time:" field, the "End Time:" field is empty. Below the "End Time:" field, there is a horizontal line. Below the horizontal line, the text "Table Properties:" is displayed.</p></div>

Attribute	Description
DisplayType	<p>Specify the Calendar display type.</p> <ul style="list-style-type: none"> Month - A 30 day calendar month view.  Day - Single day view.  Inline - Display events that occur from the start date of the calendar through the end date.  MonthUpWithEvent - Displays a month calendar. Events appear to the right of the calendar when you click on a day. 

Here is a sample of the Event Calendar syntax.

Event Calendar Sample

```
<tr>
  <td>
    <CFModule template="#request.ecm.AppPath#ecmEvtCalendar.cfm" calendar_id="24"
    DisplayType="monthupwiththeevent">
  </td>
</tr>
```

Form Function

The ecmFormBlock function displays an HTML form content block. For more information about HTML forms, see the **Ektron CMS400.NET** Administrator manual chapter "Working with HTML Forms."

IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the Ektron CMS400.NET Administrator manual chapter "Managing Library Assets", section "Updating Default Template for Multiple Quicklinks."

Here is the format for the form custom function

```
<CFMODULE template="#request.ecm.AppPath#ecmFormBlock.cfm" ID=" " >
```

This function is basically the same as the Form function in ASP. See ["Form Function" on page 485](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion example is below.

Here is an example of a form content block in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <CFModule template="#request.ecm.AppPath#ecmFormBlock" form_id="1">
  </td>
</tr>
```

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or an content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function

```
<CFMODULE template="#request.ecm.AppPath#ecmMultiPurpose.cfm" DefaultID=" " >
```

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

```
http://localhost/cms400demo/index.asp?id=2
```

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

`http://localhost/cms400demo/index.asp?ekfrm=2`

NOTE For more information about passing the id value as a url parameter, see ["Dynamic Content Block" on page 504](#).

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls `ecmContentBlock` and passes the `DefaultID` value as the id argument. If the `DefaultID` value is zero (0), `ecmContentBlock` is not called.

Metadata Function

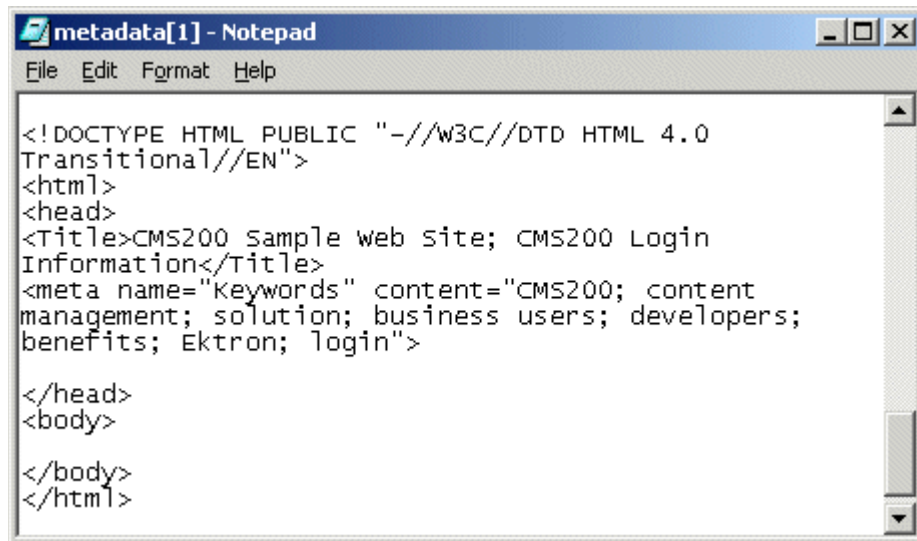
The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS400.NET Web site. Inserting this custom tag will allow you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom tag looks like.

```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="" >
```

This function is basically the same as the Metadata function in ASP. See ["Metadata Function" on page 488](#) for details.

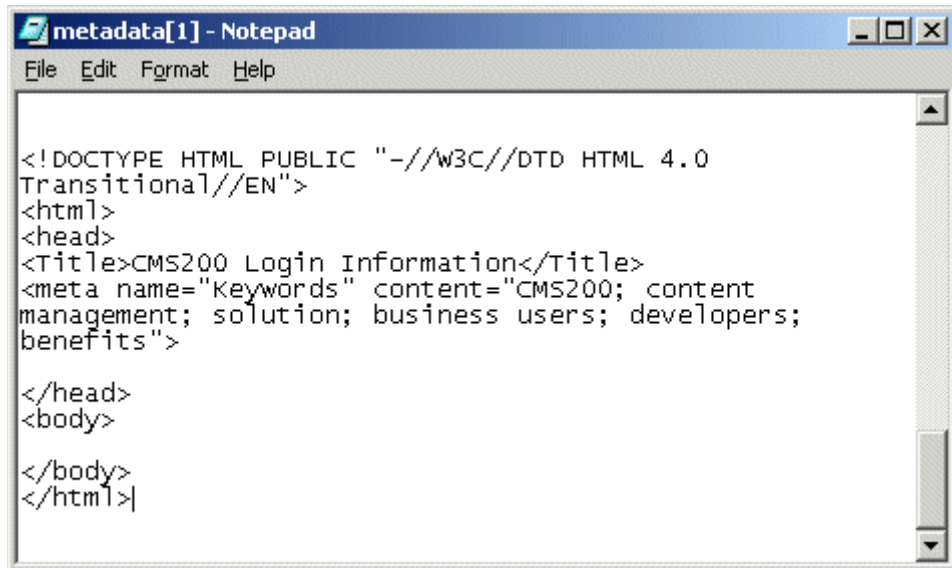
The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The ColdFusion examples are below.

```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="[1][2]">
```



```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm"
```

```
ItemList="[1;title][2;keywords]>
```



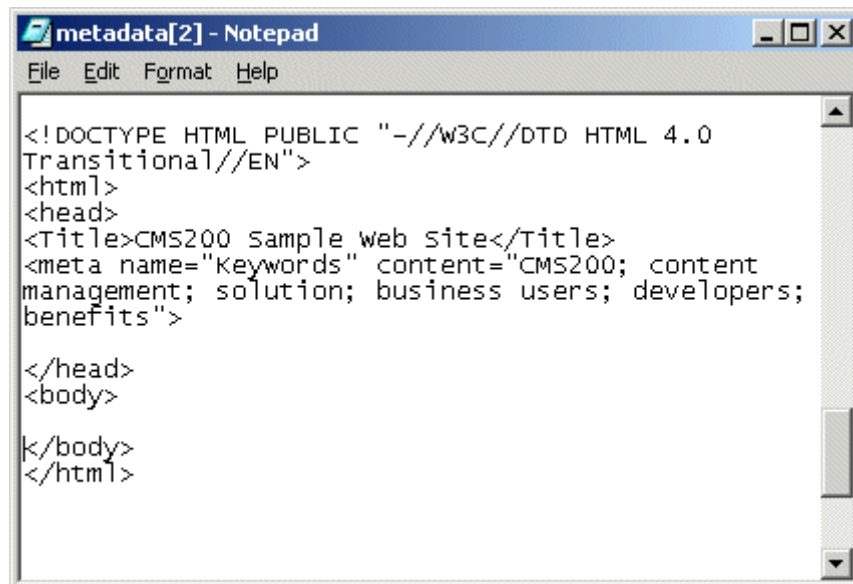
```
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>
```

```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm"
ItemList="[1][2;title,keywords]>
```



```
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>
```

DHTML Menus

Ektron CMS400.NET's Menu feature lets users create and maintain a dropdown menu system for your Web site. The Menu feature lets you display menu options that link to content blocks, library assets, external hyperlinks and submenus.

To learn more about menus, please refer to the “Menu Feature” chapter of the **Ektron CMS400.NET** User Manual.

Additional Documentation in Ektron Knowledge Base

This chapter explains how to implement standard menus that ship with **Ektron CMS400.NET**. Ektron's Web site also provides Knowledge Base articles that explain how to extend the standard menu features.

Displaying a Menu on a Web Page

GenerateMenu and DropMenu

IMPORTANT! The menu functions described below are only one example of how to implement menus. You can implement your own menu system using JavaScript.

Use the `GenerateNextGenMenu` and `DropNextGenMenu` custom functions to display a DHTML menu on a Web page. Place `GenerateNextGenMenu` once on the page to provide an infrastructure for `DropNextGenMenu`. Then, place `DropNextGenMenu` as needed to implement individual menus.

Follow these steps to display a menu on a Web template.

1. Place the following code directly after the page's `<body>` tag.

```
<CFMODULE Ids="menu id number"
template="#request.ecm.AppPath#ecmDHTML_GenerateNextGenMenu.cfm">
```

For example:

```
<CFMODULE Ids="1" template="#request.ecm.AppPath#ecmDHTML_GenerateNextGenMenu.cfm">
```

To display more than one menu, separate each id number with a comma. For example:

```
<CFMODULE Ids="1,2,3" template="#request.ecm.AppPath#ecmDHTML_GenerateNextGenMenu.cfm">
```

2. If the user clicks *text* to display the menu, place the following code within the template file where the text would appear.

```
<CFMODULE MenuId="menu id number" Title="text user clicks to display menu"
template="#request.ecm.AppPath#ecmDHTML_DropNextGenMenu.cfm">
```

For example:

```
<CFMODULE MenuId="1" Title="Products"
template="#request.ecm.AppPath#ecmDHTML_DropNextGenMenu.cfm">
```

If the user clicks *a graphic* to invoke the menu, use this syntax:

```
<CFMODULE MenuId="menu id number" Title='<img src=graphic file name>'
template="#request.ecm.AppPath#ecmDHTML_DropNextGenMenu.cfm">
```

For example:

```
<CFMODULE MenuId="1" Title='<img src=mymenugraphic.jpg>'
template="#request.ecm.AppPath#ecmDHTML_DropNextGenMenu.cfm">
```

Customizing the Menu's Appearance

You can customize a menu's appearance, such as the font style, color, and size. So, you have the flexibility to coordinate a menu with the design your Web site.

To customize a menu in **Ektron CMS400.NET**, you can change the `ek_Menu.js` file.

`ek_Menu.js`

You can change the look and feel of a menu by changing the `ek_menu.js` file. If you used the default settings when installing **Ektron CMS400.NET**, this file is in `webroot/cms400demo/Workarea/java/`. For a description of the `ek_menu.js` file, see ["ek_Menu.js" on page 496](#).

XML Menu Function

The `ecmGetMenuXML` function wraps XML around the properties of a menu created in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the menu. The `ecmGetMenuXML` function allows you to easily define how the menu data appears on the Web page.

```
<CFMODULE id="" template="#request.ecm.AppPath#ecmGetMenuXML.cfm">
```

This function is basically the same as the XML Menu function in ASP. See ["XML Menu" on page 499](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it. The Coldfusion example is below.

```
<CFMODULE id="1" template="#request.ecm.AppPath#ecmGetMenuXML.cfm">
```

Language Select Function

This function lets a visitor to your Web site select a language in which to view the site. Here is what the function looks like when published on a Web page.



Shown here is the format of the `ecmLanguageSelect` function.

```
<CFMODULE template="#request.ecm.AppPath#ecmLanguageSelect.cfm">
```

This function is basically the same as the Show Site Language function in ASP. See ["Show Site Language Function" on page 494](#) for details.

The only difference between the Coldfusion function and the ASP one is the syntax for implementing it.

Custom PHP Functions

Ektron CMS400.NET maintains and displays content via custom Ektron PHP functions and ComObjects. For the custom PHP functions to work properly, do the following:

- Make sure the `php_soap.dll` file is in the `c:\php` folder on the server where you installed PHP.
- Add the following line to the `php.ini` file.
`extension=php_soap.dll`
The `php.ini` file is located in `C:\Windows\php.ini`.
- In the `ApplicationAPI.php` file, set the `WSDLpath`. The `applicationAPI.php` file is located by default in:
`http://localhost/cms400demo/WorkArea/platforms/PHP/applicationAPI.php`

IMPORTANT!

`ContentWS.asmx` is the web services (WSDL) file that the custom PHP functions use to talk to **Ektron CMS400.NET**. The `WSDLpath` located in the `applicationAPI.php` file must be set to location of the WSDL file. The default `WSDLpath` is `http://localhost/cms400demo/WorkArea/WebServices/ContentWS.asmx`. If you have changed the location of `ContentWS.asmx` you need to update the `WSDLpath`. For example, if your site is `www.myelektronsite.com`, the `WSDLpath` would be `http://www.myelektronsite.com/WorkArea/WebServices/ContentWS.asmx`.

- Add the following line to the top of each template.
`<?php include 'applicationAPI.php'; ?>`
This file calls the Ektron custom functions file to allow the page to function properly. It *must* appear at the top of the template file.
- If you have upgraded your Ektron CMS400.NET from an Ektron CMS300 version, add the following line between the head tags for each template.

IMPORTANT!

This only needs to be done if you are upgrading from CMS300 to CMS400.NET

```
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
```

To customize you Ektron CMS400.NET application, you need to be familiar with these functions.



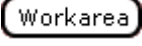
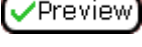

Function	Description	More Information
Calendar Functions		
Event Calendar	Displays a month view event calendar	"Event Calendar Function" on page 534

Function	Description	More Information
Collection Functions		
Collection	Displays links generated by collections	"Collection Function" on page 529
CollectionXML	Creates XML for a collection by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML Collection Function" on page 532
Random Content	Randomly displays the content of a collection item	"Random Content Function" on page 533
Random Summary	Randomly displays the summary of a collection item	"Random Summary Function" on page 533
RSS Collection	Creates an RSS feed out of the items in a collection.	"RSS Collection Function" on page 533
Content Functions		
Add Content	Allows a user to add content to Ektron CMS400.NET programmatically.	"Add Content" on page 525
Content Block	Displays and manages a content block	"Content Block Function" on page 524
Form	Displays a content block associated with a form	"Form Function" on page 536
Multipurpose	Displays either a content block or a form	"MultiPurpose Function" on page 537
List Summary Functions		
Archive ListSummary	Displays summaries of archived content blocks	"Archive List Summary Function" on page 528
List Summary	Displays a list of content block summaries	"List Summary Function" on page 527

Function	Description	More Information
List Summary XML	Creates XML for a list summary by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML List Summary Function" on page 529
RSS List Summary	Creates an RSS feed out of the items in a list summary.	"RSS List Summary Function" on page 529
Single Summary	Displays the summary for a content block	"Single Summary Function" on page 528
Login Functions		
Login	Adds the login or logout button	"Login Function" on page 524
Menu Functions		
DropHTMLMenu	Used to display a DHTML menu	"DHTML Menus" on page 539
GenerateMenu	Works with DropHTML menu to display DHTML menus	"DHTML Menus" on page 539
XML Menu	Creates XML for a menu by wrapping tags around all of the menu items properties.	"XML Menu Function" on page 541
Metadata Functions		
Metadata	Displays the metadata for the current Web page	"Metadata Function" on page 537
Search Functions		
Search	Inserts a text box with a search button	"Search Function" on page 526
Search Display	Creates an area to display search results	"Search Display Function" on page 527
Language Functions		
Language Select	Lets a site visitor select a language in which to view site.	"Language Select Function" on page 541

Login Function

This function is responsible for displaying the following buttons on the Web page.

Button	Name	Description
	Login	When user is not logged into Ektron CMS400.NET, this button appears. Click this button to let the user log in to the application.
	Logout	Once logged in, this button replaces the login button to allow the user to log out from the Ektron CMS400.NET Web site.
	Workarea	When logged in, this button appears under the logout button allowing the user to access their workarea.
	Preview On	Lets the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off preview mode.

This function is basically the same as the Login function in ASP. See "[Login Function](#)" on page 414 for details. The only difference is the syntax used to implement the function. This is listed below.

Login Function Example

```
<tr>
  <td>
    <?php ecmlogin(); ?>
  </td>
</tr>
```

Content Block Function

The `ecmContentBlock` and `ecmContentBlockEX` functions are used to display a content block on an Ektron CMS400.NET Web page.

Shown here is the format for the `ecmContentBlock` function:

```
<?php ecmContentBlock (ID) ?>
```

Here is the format for the `ecmContentBlockEX` function:

```
<?php ecmContentBlockEx (ID, "XSLT", Override) ?>
```

NOTE As an alternative, you can use the [multipurpose function](#), which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 537](#).

This function is basically the same as the Content Block function in ASP. See ["Content Block Function" on page 416](#) for details.

The only differences between the PHP function and the ASP one are the examples of implementing it. The PHP examples are below.

Static Content Block

```
<tr>
  <td>
    <?php  ecmContentBlock(1) ?>
  </td>
</tr>
```

Dynamic Content Block

```
<tr>
  <td>
    <?php  ecmContentBlock($id) ?>
  </td>
</tr>
```

XML Content Block

```
<tr>
  <td>
    <?php ecmContentBlockEx (13, "", 1) ?>
  </td>
</tr>
```

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS400.NET without the need of logging in.

Here is the format for the add content function:

```
<? AddContent      ("Username"
                    "Password",
                    "Domain",
                    'content_title',
                    "content_comment",
                    "ContentHtml",
                    "SummaryHTML",
                    "ContentLanguage",
                    "FolderID",
                    "GoLive",
                    "EndDate",
                    "MetaInfo");           ?>
```

This function is basically the same as the Add Content function in ASP. See ["Add Content" on page 424](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

Add Content Example

The following is an example of the add content function in a template:.

```
<tr>
  <td>
    <? AddContent ("admin", "admin", "", "Content Title", "Comment",
  "This is the <b>body</b> of the content.", "This is the <i>summary</i>
  for the content.", "1033", "0", "", "", ""); ?>
  </td>
</tr>
```

Search Function

The search function is used to display a text box with a search button on the template. When seen on the template on the Web page, a user can enter text and click the search button to search through all the content on the Web site.

Ektron has created two search functions:

- ecmSearch
- ecmSearchNoFormTag

Both functions have the same parameters and perform the same search. The only difference is ecmSearch automatically inserts form tags, ecmSearchNoFormTag does not.

The following is an example of form tags.

```
<form name="ecmsearch" method="post" action='search-
display.asp'>
```

```
</form>
```

When using ecmSearch, form tags are added automatically. When using ecmSearchNoFormTag you need to add the tags manually

Here is the format for the search function. You can interchange ecmSearch and ecmSearchNoFormTag when using this function.

```
<?php ecmSearch( "StartingFolder",
Recursive,
"targetpage",
textbox,
MaxCharacters,
"ButtonImgSrc",
"ButtonText",
"FontFace",
"FontColor",
"FontSize",
```

```

        Horizontal,
        "Spare" );
?>

```

This function is basically the same as the Search function in ASP. See ["Search Functions" on page 428](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

Search Function Example

```

<TR>
  <TD>
    <?php ecmSearch("\Marketing",1,"search-
display.php",25,200,"","Search","Verdana","#808080","2",0,""); ?>
  </TD>
</TR>

```

Search Display Function

The search display custom function created the area where all the search results show after the search has been executed.

Here is the format of the ecmSearchDisplay function:

```

<?php ecmSearchDisplay( MaxNumbers,"StyleInfo", ShowDate ); ?>

```

This function is basically the same as the Search Display function in ASP. See ["Search Display Function" on page 438](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

Search Display Function Example

```

<tr>
  <td>
    <?php ecmSearchDisplay( 0,"",1); ?>
  </td>
</tr>

```

List Summary Function

The list summary function is used to display the list of content summaries on a Web page. Here is the format of the list summary function.

```

<?php ecmListSummary ( "Folder",
    Recursive,
    ShowSummary,
    "StyleInfo",
    "OrderBy",
    "ShowInfo",
    MaxNumber,
    "Spare 1",
    "Spare2" );
?>

```

This function is basically the same as the List Summary function in ASP. See ["List Summary Function" on page 442](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

List Summary Example

```
<TR>
  <TD>
    <?PHP ecmListSummary ("Marketing\News",0,1,"",
      "Title,asc","DateModified",0,"",""); ?>
  </TD>
</TR>
```

Archive List Summary Function

The Archive list summary function displays on a Web page a list of content summaries for archived content blocks. For example, you might have a current news page and an archived news page. Use the List Summary Function to display current news, and this function to display older news stories.

On the Content Block **schedule** screen, the user must set an **End Date** then choose the Archive and Display option to have the content blocks appear via this function.

This function has the same parameters as the List Summary Function. See ["List Summary Function" on page 527](#) for details.

Single Summary Function

The single summary function is used to display the summary of one content block in the Ektron CMS400.NET Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary function:

```
<?php ecmSingleSummary( "ContentID",
  ShowSummary,
  "StyleInfo",
  "ShowInfo",
  "Spare1");
?>
```

This function is basically the same as the Single Summary function in ASP. See ["Single Summary Function" on page 456](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

Single Summary Example

```
<TR>
  <TD>
    <?PHP ecmSingleSummary ("14",1,"","DateModified",""); ?>
  </TD>
</TR>
```

XML List Summary Function

The XML List Summary function wraps XML around the properties of a list of content, created as a list summary in the Ektron CMS400.NET Workarea. Once created, you can use the XML to display the list summary information. The `ecmListSummaryXML` function allows you to easily define how the list summary data appears on the Web page.

```
<?php ecmListSummaryXML ("Folder", Recursive,
"OrderBy", MaxNumber, "ObjType", "SummaryType"); ?>
```

This function is basically the same as the List Summary XML function in ASP. See ["XML List Summary Function" on page 450](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

```
<?php ecmListSummaryXML("\books\\",0,"Title,asc",0,"folder",""); ?>
```

RSS List Summary Function

The RSS summary function is used to create an XML list of content summaries for RSS Syndication. Shown here is the format of the RSS list summary function:

```
<?php ecmRssSummary ("Folder", Recursive, "OrderBy", MaxNumber); ?>
```

This function is basically the same as the RSS Summary function in ASP. See ["RSS Summary Function" on page 460](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

```
<?php ecmRssSummary("\CMS400Demo\\", 0, "Title,asc", 0); ?>
```

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS400.NET Workarea. The `ecmCollection` function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the `ecmCollection` function.

```
<?PHP ecmCollection( "id",
"displayFunction";
?>
```

This function is basically the same as the Collection function in ASP. See ["Collection Function" on page 465](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

Collection Examples

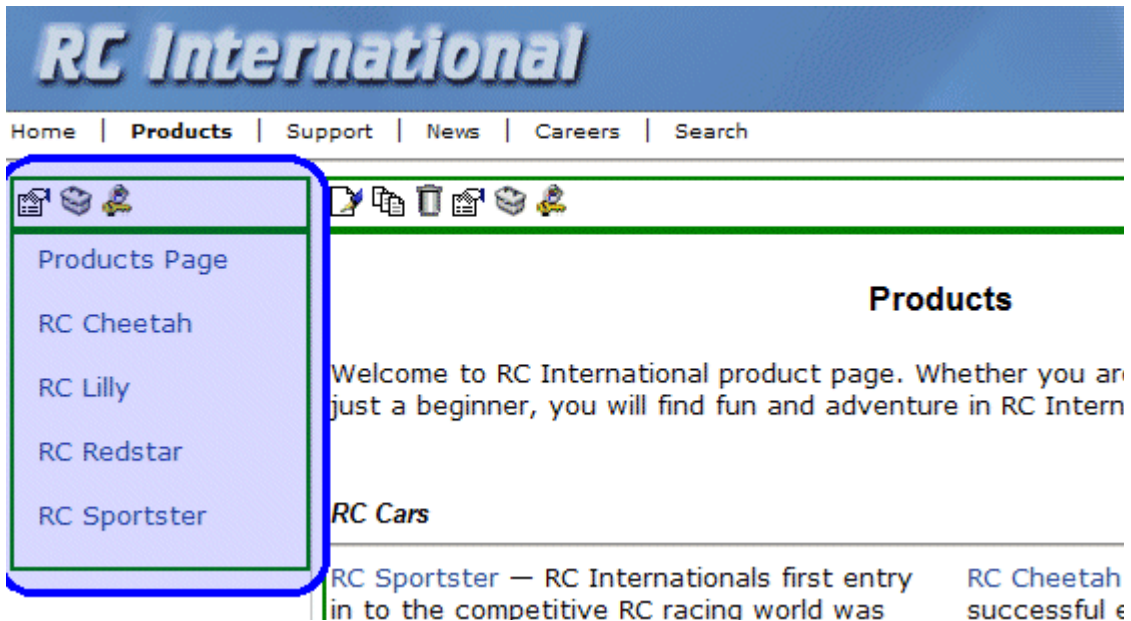
Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS400.NET sample Web site.

```
<?php ecmCollection(1, "ecmNavigation") ?>
```

In this example, the collection with an ID=1 is displayed, and the function "ecmNavigation" is used to define how the collection data is displayed on the Web page.

Here is how the collection would appear on the Web page.



The format of the collection on the Web page depends on the displayFunction that is used.

Below is the source code for the display function "ecmNavigation".

```

Function ecmNavigation($cInfo){
    global $html, $info;
    $html = "<table border=\"0\" cellspacing=\"0\" cellpadding=\"0\"
        width=\"100%\">";
    $mycount = $cInfo->count();
    for ($iloop=0; $iloop < $mycount; ++$iloop) {
        $info = $cInfo->Item($iloop+1);
        $html = $html."<tr><td>&nbsp;&nbsp;&nbsp;<a href=\"";
        $html = $html.$info->Item("ContentLinks");
        $html = $html.\">\";
        $html = $html.$info->Item("ContentTitle");
        $html = $html.\"</a></td></tr><tr><td>&nbsp;&nbsp;&\"</td></tr>\";
    }
    $html = $html.\"</table>\";
    return $html;
}

```

As you can see, the ecmNavigation function is a simple PHP function that creates a table of the links that belong to the collection specified.

The ecmNavigation function also displays the title of the content blocks by using the "info("ContentTitle")) building block.

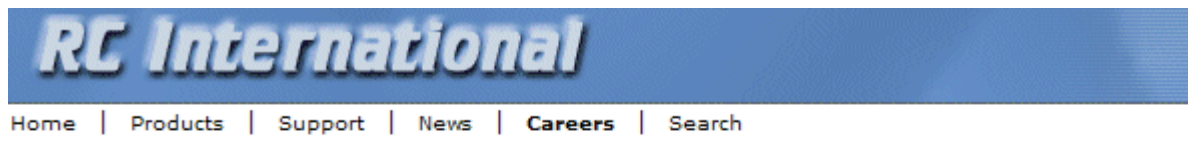
Example Using the ecmTeaser Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS400.NET sample Web site.

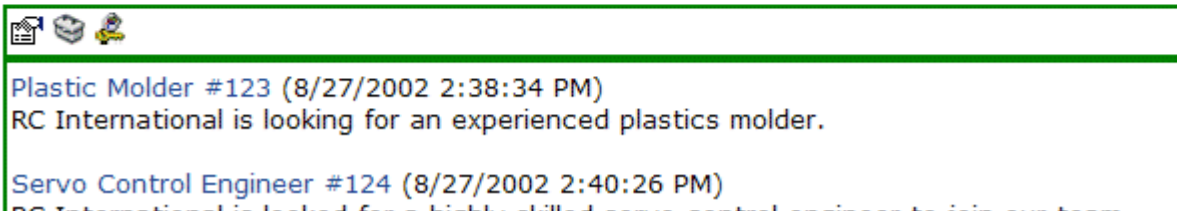
```
<?php ecmCollection (2, "MyDisplayFunction"); ?>
```

In this example, the collection with an ID=2 is displayed, and the function "ecmTeaser" is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



Employment Opportunities



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function "ecmTeaser"

```

Function ecmTeaser($cInfo){
    global $html, $info;
    $html = "<table border=\"0\" cellspacing=\"0\" cellpadding=\"0\" width=\"100%\">";
    $mycount = $cInfo->count();
    for ($i=0; $i < $mycount; ++$i) {
        $info = $cInfo->Item($i+1);
        $html = $html."<tr><td><a href=\"";
        $html = $html.$info->Item("ContentLinks");
        $html = $html.\" \">";
        $html = $html.$info->Item("ContentTitle");
        $html = $html."</a>&nbsp;";
        $html = $html.date("m/d/Y g:i:s A", $info->Item("DateModified"));
        $html = $html.\"</td></tr><tr><td>";
        $html = $html.$info->Item("ContentTeaser");
        $html = $html.\"</td></tr><tr><td>&nbsp;</td></tr>";
    }
    $html = $html."</table>";
    return $html;
}

```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmTeaser function also displays the:

- Content Title
- Date Modified
- Content Teaser

Collection Variables

See ["Collection Attributes" on page 468](#)

DisplayFunction

As explained earlier, the display function is a function that you create that will define how the collection data will be displayed on the Web page.

By using simple PHP scripting, you can create your own display functions, or use the three included functions.

XML Collection Function

The XML collection function wraps XML around the parameters of a list of content that's created as a collection in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the collection information. The ecmCollectionXML function allows you to easily define how the collection data appears on the Web page.

Shown here is the format of the ecmCollection function.

```
<?php ecmCollectionXML (id, GetHtml); ?>
```

This function is basically the same as the XML Collection function in ASP. See ["XML Collection Function" on page 471](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

```
<?php ecmCollectionXML (3,0); ?>
```

RSS Collection Function

The RSS collection function is used to create an XML list of links that were created as a collection in the Ektron CMS400.NET Workarea and display them in RSS Syndication.

Shown here is the PHP format of the ecmRssCollection function.

<?php ecmRssCollection (id); ?>

This function is basically the same as the RSS Collection function in ASP. See ["RSS Collection Function" on page 469](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it.

Random Content Function

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

<?php ShowRandomContent (id); ?>

This function is basically the same as the Random Content function in ASP. See ["Random Content" on page 476](#) for details.

Random Summary Function

Similar to the random content function, the random summary function is used to randomly display the summary of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

<?php ShowRandomSummary (id, XsltID); ?>

This function is basically the same as the Random Summary function in ASP. See ["Random Summary" on page 477](#) for details.

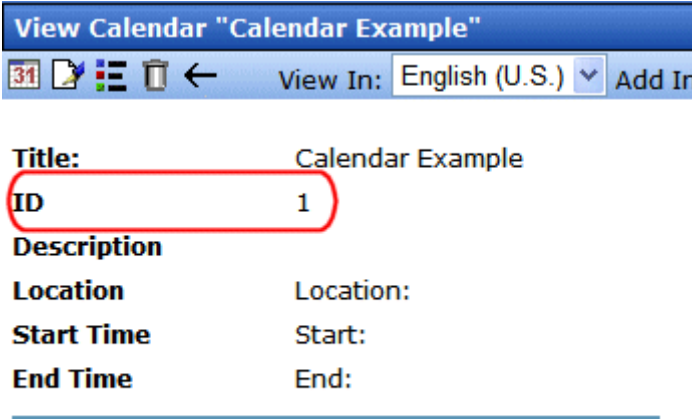
Event Calendar Function


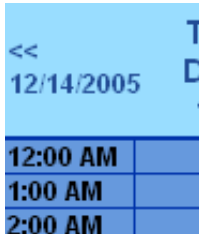
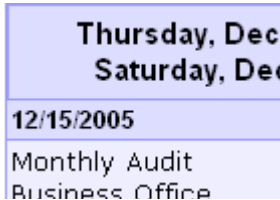

The ecmEvtCalendar function is used to display an event calendar that has been created in the Ektron CMS400.NET Workarea.

```
<?php ecmEvtCalendar(id, "Displaytype"); ?>
```

This function is basically the same as the Event Calendar function in ASP. See ["Event Calendar Function" on page 478](#) for details.

The only differences between the PHP function and the ASP one are the use of the DisplayType parameter and the syntax for implementing it. The ecmEvtCalendar function parameters are defined below.

Attribute	Description
id	<p>The "id" argument represents the ID number of the event calendar in the Workarea.</p>  <p>Title: Calendar Example</p> <p>ID 1</p> <p>Description</p> <p>Location Location:</p> <p>Start Time Start:</p> <p>End Time End:</p> <hr/> <p>Table Properties:</p>

Attribute	Description
DisplayType	<p>Specify the Calendar display type.</p> <ul style="list-style-type: none"> Month - A 30 day calendar month view.  Day - Single day view.  Inline - Display events that occur from the start date of the calendar through the end date.  MonthUpWithEvent - Displays a month calendar. Events appear to the right of the calendar when you click on a day. 

Event Calendar Sample

Here is an example of the event Calendar in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <?php ecmEvtCalendar(24, "MonthUpWithEvent"); ?>
  </td>
</tr>
```

Form Function

The `ecmFormBlock` and `ecmFormBlockNoFormTag` functions display an HTML form content block. Both functions have the same parameters and display the form the same way. The only difference is `ecmFormBlock` automatically inserts form tags, `ecmFormBlockNoFormTag` does not.

The following is an example of form tags.

```
<form name="myform" action="#" onsubmit="return
EkFmValidate(this);" method="post">

</form>
```

When using `ecmFormBlock`, form tags are added automatically. When using `ecmFormBlockNoFormTag`, form tags need to be added manually.

For more information about HTML forms, see the **Ektron CMS400.NET Administrator manual** chapter "Working with HTML Forms."

IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the **Ektron CMS400.NET Administrator manual** chapter "Managing Library Assets", section "Updating Default Template for Multiple Quicklinks."

Here is the format for the form custom function

```
<?php ecmFormBlock(id); ?>
<?php ecmFormBlockNoForTag(id); ?>
```

This function is basically the same as the Form function in ASP. See ["Form Function" on page 485](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

```
<tr>
  <td>
    <?php ecmFormBlock(1); ?>
  </td>
</tr>
```

NOTE As an alternative, you can use the [multipurpose function](#), which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 537](#).

MultiPurpose Function

The `ecmMultiPurpose` function displays either a content block or a content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function

`<?php ecmMultiPurpose(ID); ?>`

To pass in a standard content block, pass the `id` value in the `url` parameter, as illustrated below.

```
http://localhost/cms400demo/index.asp?id=2
```

To pass in a Form content block, pass the `ekfrm` value in the `url` parameter, as illustrated below.

```
http://localhost/cms400demo/index.asp?ekfrm=2
```

NOTE For more information about passing the `id` value as a `url` parameter, see ["Dynamic Content Block" on page 525](#).

If the `id` value or `ekfrm` value does not exist, a blank template appears.

If the `url` parameter is not passed in or recognized, the function calls `ecmContentBlock` and passes the `DefaultID` value as the `id` argument. If the `DefaultID` value is zero (0), `ecmContentBlock` is not called.

Metadata Function

The metadata function is used to display the metadata for one or more content blocks in the Ektron CMS400.NET Web site. Inserting this custom function allows you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom function looks like.:

```
<?php ecmMetaData ("ItemList","Spare1"); ?>
```

This function is basically the same as the Metadata function in ASP. See ["Metadata Function" on page 488](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP examples are below.

To include both content blocks' metadata in the web page, an `ecmMetadata` call would look like this.

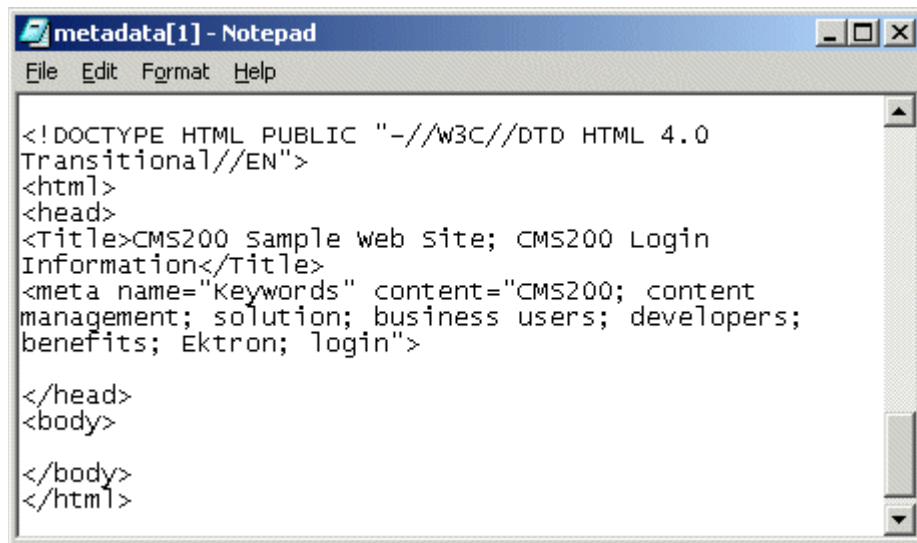
```
<html>
<head>
    <?php ecmMetadata ("[1][2;Title,Content-Type]", ""); ?>
    (rest of head)
```

```
</head>
<body>
    (rest of web page document)
</body>
</html>
```

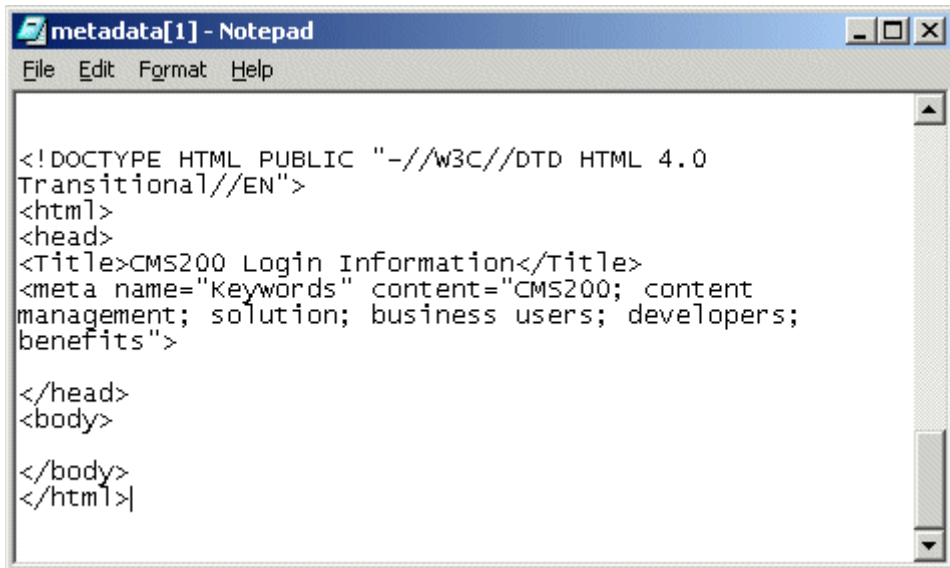
In the above example the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). Once the Web page is viewed the Web page source will contain the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
    product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

```
<?PHP ecmMetaData ("[1][2]","") ?>
```



```
<?PHP ecmMetaData ("[1;title][2;keywords]","") ?>
```



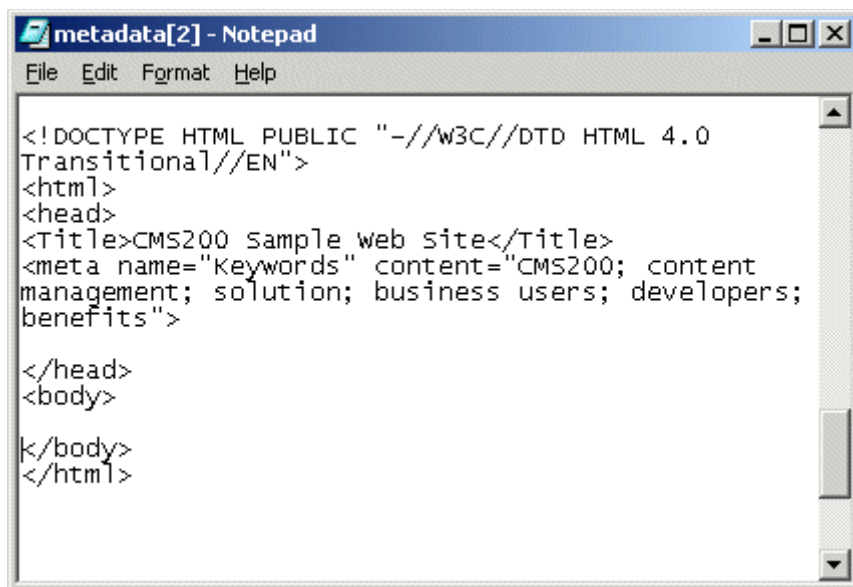
```
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>
```

```
<?PHP ecmMetaData ("[1][2;title,keywords]","") ?>
```



```
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>
```

DHTML Menus

Ektron CMS400.NET's Menu feature lets users create and maintain a dropdown menu system for your Web site. The Menu feature lets you display menu options that link to content blocks, library assets, external hyperlinks and submenus.

To learn more about menus, please refer to the "Menu Feature" chapter of the **Ektron CMS400.NET** User Manual.

Additional Documentation in Ektron Knowledge Base

This chapter explains how to implement standard menus that ship with **Ektron CMS400.NET**. Ektron's Web site also provides Knowledge Base articles that explain how to extend the standard menu features.

Displaying a Menu on a Web Page

GenerateMenu and DropMenu

IMPORTANT!

The menu functions described below are only one example of how to implement menus. You can implement your own menu system using JavaScript.

Use the `GenerateNextGenMenu` and `DropNextGenMenu` custom functions to display a DHTML menu on a Web page. Place `GenerateNextGenMenu` once on the page to provide an infrastructure for `DropNextGenMenu`. Then, place `DropNextGenMenu` as needed to implement individual menus.

Follow these steps to display a menu on a Web template.

1. Place the following code directly after the page's `<body>` tag.

```
<?php ecmDHTML_GenerateNextGenMenu("menu id number"); ?>
```

For example:

```
<?php ecmDHTML_GenerateNextGenMenu("1"); ?>
```

To display more than one menu, separate each id number with a comma. For example:

```
<?php ecmDHTML_GenerateNextGenMenu("1,2,3"); ?>
```

2. If the user clicks *text* to display the menu, place the following code within the template file where the text would appear.

```
<?php ecmDHTML_DropNextGenMenu menu ID number,"text user clicks to display menu"; ?>
```

For example:

```
<?php ecmDHTML_DropNextGenMenu 1,"MyMenu1"; ?>
```

If the user clicks *a graphic* to invoke the menu, use this syntax:

```
<?php ecmDHTML_DropNextGenMenu menu ID number, '<img src=graphic file name>'; ?>
```

For example:

```
<?php ecmDHTML_DropNextGenMenu 1,'<img src=mymenugraphic.jpg>'; ?>
```

Customizing the Menu's Appearance

You can customize a menu's appearance, such as the font style, color, and size. So, you have the flexibility to coordinate a menu with the design your Web site.

To customize a menu in **Ektron CMS400.NET**, you can change the `ek_Menu.js` file.

ek_Menu.js

You can change the look and feel of a menu by changing the `ek_menu.js` file. If you used the default settings when installing **Ektron CMS400.NET**, this file is in

`webroot/cms400demo/Workarea/java/`. For a description of the `ek_menu.js` file, see ["ek_Menu.js" on page 496](#).

XML Menu Function

The `ecmGetMenuXML` function wraps XML around the properties of a menu created in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the menu. The `ecmGetMenuXML` function allows you to easily define how the menu data appears on the Web page.

`<?php ecmGetMenuXML($Id); ?>`

This function is basically the same as the XML Menu function in ASP. See ["XML Menu" on page 499](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it. The PHP example is below.

```
<?php ecmGetMenuXML(1); ?>
```

Language Select Function

This function lets a visitor to your Web site select a language in which to view the site. Here is what the function looks like when published on a Web page.



Shown here is the format of the `ecmLanguageSelect` function.

`<?php ecmLanguageSelect(); ?>`

This function is basically the same as the Show Site Language function in ASP. See ["Show Site Language Function" on page 494](#) for details.

The only difference between the PHP function and the ASP one is the syntax for implementing it.

Custom API

As you have seen, the formatting for the custom functions used in Ektron CMS400.NET for PHP are defined in the file `ApplicationAPI.php`. As a developer, you may make changes to this file to further enhance how the custom functions appear on your Web site.

However, Ektron recommends creating a copy of the file, and renaming it to an arbitrary name, for example, `MyCustomAPI.php`, and including that file when creating Ektron CMS400.NET templates. This ensures that your custom API will not be overwritten if you upgrade to another version of Ektron CMS400.NET.

Custom JSP Functions

Ektron CMS400.NET maintains and displays content via custom Ektron JSP functions. For the custom JSP functions to work properly, do the following:

- In the `applicationapi.jsp` file, set the `WSSoap_address`. The `applicationapi.jsp` file is located by default in:
`http://localhost/cms400demo/WorkArea/platforms/JSP/applicationapi.jsp`

IMPORTANT!

`ContentWS.asmx` is the web services file that the custom JSP functions use to talk to **Ektron CMS400.NET**. The `WSSoap_address` located in the `applicationapi.jsp` file must be set to location of the Web services file. The default `WSSoap_address` is `http://localhost/cms400demo/WorkArea/WebServices/ContentWS.asmx`. If you have changed the location of `ContentWS.asmx` you need to update the `WSSoap_address`. For example, if your site is `www.myelektronsite.com`, the `WSSoap_address` would be `http://www.myelektronsite.com/WorkArea/WebServices/ContentWS.asmx`.

- Add the following line to the top of each template.

```
<%@ include file="applicationapi.jsp" %>
```

This file calls the Ektron custom functions file to allow the page to function properly. It *must* appear at the top of the template file.
- Make sure the `contentws.jar` file is included in your JSP environment
- Ektron recommends running Axis when using a JSP environment. We have tested on TomCat and JRUN with Axis v1.2.1. You can download Axis from <http://ws.apache.org/axis/java/>.

To customize you Ektron CMS400.NET application, you need to be familiar with these functions.

Function	Description	More Information
Calendar Functions		
Calendar	Displays a calendar with different display types. For example, a daily view calendar. For more information on the types of calendars that can be displayed, see "DisplayType" on page 482	"Calendar Function" on page 545
Event Calendar	Displays a month view event calendar	"Event Calendar Function" on page 546
Collection Functions		

Function	Description	More Information
Collection	Displays links generated by collections	"Collection Function" on page 547
CollectionXML	Creates XML for a collection by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML Collection Function" on page 547
Random Content	Randomly displays the content of a collection item	"Random Content Function" on page 548
Random Summary	Randomly displays the summary of a collection item	"Random Summary Function" on page 548
RSS Collection	Creates an RSS feed out of the items in a collection.	"RSS Collection Function" on page 548
Content Functions		
Add Content	Allows a user to add content to Ektron CMS400.NET programmatically.	"Add Content" on page 549
Content Block	Displays and manages a content block	"Content Block Function" on page 549
Form	Displays a content block associated with a form	"Form Function" on page 550
Language Functions		
Language Select	Lets a site visitor select a language in which to view site.	"Language Select Function" on page 552
Login Functions		
Login	Adds the login or logout button	"Login Function" on page 552
Menu Functions		
DropHTMLMenu	Used to display a DHTML menu	"DHTML Menus" on page 553

Function	Description	More Information
GenerateMenu	Works with DropHTMLMenu to display DHTML menus.	"DHTML Menus" on page 553
XML Menu	Creates XML for a menu by wrapping tags around all of the menu items properties.	"XML Menu Function" on page 554
Metadata Functions		
Metadata	Displays the metadata for the current Web page.	"Metadata Function" on page 555
Search Functions		
Search	Inserts a text box with a search button.	"Search Function" on page 557
Search Display	Creates an area to display search results.	"Search Display Function" on page 558
Summary Functions		
List Summary	Displays a list of content block summaries.	"List Summary Function" on page 559
List Summary XML	Creates XML for a list summary by wrapping tags around all of the collection items and their properties. This allows you to use an XSLT to define how the XML collection is displayed.	"XML List Summary Function" on page 559
RSS List Summary	Creates an RSS feed out of the items in a list summary.	"RSS List Summary Function" on page 559
Single Summary	Displays the summary for a content block.	"Single Summary Function" on page 560

Calendar Function

The ecmCalendar function displays a calendar that has been created in Ektron CMS400.NET. Here is the format for the event calendar custom function.

```
<%= ecmCalendar (int calendarID, String "displayType") %>
```

NOTE

While you can display a month view calendar with `ecmCalendar`, you can also use `ecmEvtCalendar` to display a month view calendar. For more information, see ["Event Calendar Function" on page 546](#)

This function is basically the same as the `Calendar` function in ASP. See ["Calendar Function" on page 481](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it.

Calendar Sample

Here is an example of the `Calendar` function in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <%= ecmEvtCalendar(24, "MonthUpWithEvent") %>
  </td>
</tr>
```

Event Calendar Function

The `ecmEvtCalendar` function is used to display an event calendar that has been created in the Ektron CMS400.NET Workarea.

`<%= ecmEvtCalendar(int calendar id) %>`

This function is basically the same as the `Event Calendar` function in ASP. See ["Event Calendar Function" on page 478](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it.

Event Calendar Sample

Here is an example of the event `Calendar` in an Ektron CMS400.NET template.

```
<tr>
  <td>
    <%= ecmEvtCalendar(24) %>
  </td>
</tr>
```

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS400.NET Workarea. The `ecmCollection` function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the `ecmCollection` function.

`<%= ecmCollection(int id, String "displayFunction") %>`

This function is basically the same as the `Collection` function in ASP. See ["Collection Function" on page 465](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it.

Collection Sample

Here is an example of the `Collection` function in an Ektron CMS400.NET template.

```
<tr>
  <td>

    <%= ecmCollection(24, "ecmTeaser") %>

  </td>
</tr>
```

XML Collection Function

The XML collection function wraps XML around the parameters of a list of content that's created as a collection in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the collection information. The `ecmCollectionXML` function allows you to easily define how the collection data appears on the Web page. Shown here is the format of the `ecmCollection` function.

`<%= ecmCollectionXML (int id, boolean GetHtml) %>`

This function is basically the same as the XML `Collection` function in ASP. See ["XML Collection Function" on page 471](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

```
<tr>
  <td>

    <%= ecmCollectionXML(24, True) %>

  </td>
</tr>
```

Random Content Function

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

```
<%= ecmShowRandomContent (int collectionID, int xsltId) %>
```

This function is basically the same as the Random Content function in ASP. See ["Random Content" on page 476](#) for details.

Random Summary Function

Similar to the random content function, the random summary function is used to randomly display the summary of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

```
<%= ShowRandomSummary (int collectionID) %>
```

This function is basically the same as the Random Summary function in ASP. See ["Random Summary" on page 477](#) for details.

RSS Collection Function

The RSS collection function is used to create an XML list of links that were created as a collection in the Ektron CMS400.NET Workarea and display them in RSS Syndication.

Shown here is the JSP format of the ecmRssCollection function.

```
<%= ecmRssCollection (int id) %>
```

This function is basically the same as the RSS Collection function in ASP. See ["RSS Collection Function" on page 469](#) for details.

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS400.NET without the need of logging in.

Here is the format for the add content function:

```
<%= ecmAddContent (String UserName, String Password, String Domain, String content_title,
String content_comment, String ContentHtml, String SummaryHTML, String ContentLanguage,
String FolderID, String GoLive, String EndDate, String MetaInfo) %>
```

This function is basically the same as the Add Content function in ASP. See ["Add Content" on page 424](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

Add Content Example

The following is an example of the add content function in a template:.

```
<tr>
  <td>
    <%= ecmAddContent ("admin", "admin", "", "Content Title",
"Comment", "This is the <b>body</b> of the content.", "This is the
<i>summary</i> for the content.", "1033", "0", "", "", "") %>
  </td>
</tr>
```

Content Block Function

The ecmContentBlock and ecmContentBlockEX functions are used to display a content block on an Ektron CMS400.NET Web page.

Shown here is the format for the ecmContentBlock function:

```
<%= ecmContentBlock (int id) %>
```

Here is the format for the ecmContentBlockEX function:

```
<%= ecmContentBlockEX (int id, String xslt, int OverrideId) %>
```

These function is basically the same as the Content Block functions in ASP. See ["Content Block Function" on page 416](#) for details.

The only difference between the JSP function and the ASP one is the examples of implementing it. The JSP examples are below.

Static Content Block

```
<tr>
  <td>
    <%= ecmContentBlock(1) %>
  </td>
</tr>
```

Dynamic Content Block

```
<tr>
  <td>
    <%= ecmContentBlock($id) %>
  </td>
</tr>
```

XML Content Block

```
<tr>
  <td>
    <%= ecmContentBlockEx (13, "", 1) %>
  </td>
</tr>
```

Form Function

The `ecmFormBlock` function displays an HTML form content block. This function has a parameter that lets you decide whether to have form tags added automatically or manually. The following is an example of form tags.

```
<form name="myform" action="#" onsubmit="return
EkFmValidate(this);" method="post">

</form>
```

For more information about HTML forms, see the **Ektron CMS400.NET Administrator** manual chapter “Working with HTML Forms.”

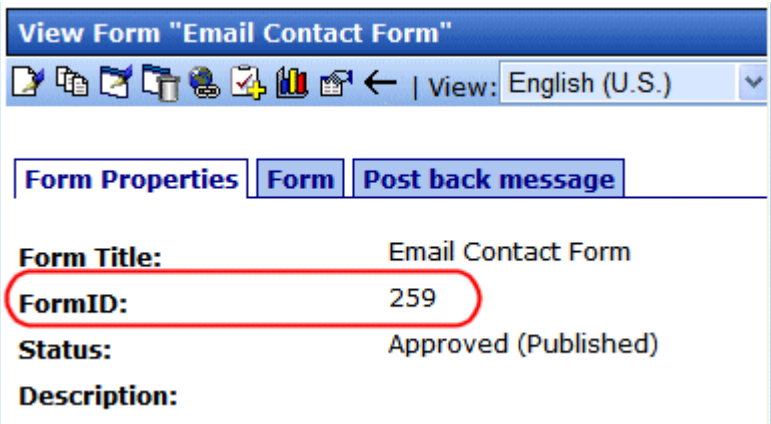
IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the **Ektron CMS400.NET Administrator** manual chapter “Managing Library Assets”, section “Updating Default Template for Multiple Quicklinks.”

Here is the format for the `ecmFormBlock` function:

`<%= ecmFormBlock (int id, boolean formTagReqd) %>`

The ecmFormBlock function attributes are explained below.

Attribute	Description
id	<p>The "id" argument represents the form's ID number, visible when viewing the form from the View Form screen.</p>  <p>The screenshot shows a web interface titled 'View Form "Email Contact Form"'. It includes a toolbar with various icons and a language dropdown set to 'English (U.S.)'. Below the toolbar are three tabs: 'Form Properties', 'Form', and 'Post back message'. The 'Form Properties' tab is active, displaying the following information:</p> <ul style="list-style-type: none"> Form Title: Email Contact Form FormID: 259 (highlighted with a red circle) Status: Approved (Published) Description:
formTagReqd	<p>Allows form tags to be added automatically.</p> <p>This is an example of form tags.</p> <pre><form name="myform" action="#" onsubmit="return EkFmValidate(this);" method="post"> </form></pre> <p>true = Add form tags automatically false = Add form tags manually</p>

This function is basically the same as the Form function in ASP. See ["Form Function" on page 485](#) for details.

The only differences between the JSP function and the ASP one are the formTagReqd parameter and the syntax for implementing it. The JSP example is below.

```
<tr>
  <td>
    <%= ecmFormBlock(1, true) %>
  </td>
</tr>
```

Language Select Function

This function lets a visitor to your Web site select a language in which to view the site. Here is what the function looks like when published on a Web page.

Site Language:

- Danish
- English (US)
- French (Standard)
- German (Standard)

Shown here is the format of the ecmLanguageSelect function.



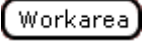
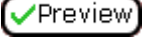
```
<%= ecmLanguageSelect() %>
```


This function is basically the same as the Show Site Language function in ASP. See ["Show Site Language Function" on page 494](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it.

Login Function

This function is responsible for displaying the following buttons on the Web page.

Button	Name	Description
	Login	When user is not logged into Ektron CMS400.NET, this button appears. Click this button to let the user log in to the application.
	Logout	Once logged in, this button replaces the login button to allow the user to log out from the Ektron CMS400.NET Web site.
	Workarea	When logged in, this button appears under the logout button allowing the user to access their workarea.
	Preview On	Lets the user to preview the entire Web site as if all checked-in content was published.

Button	Name	Description
	Preview Off	Turns off preview mode.

Shown here is the format of the ecmLogin function.

```
<%= ecmlogin() %>
```

This function is basically the same as the Login function in ASP. See "[Login Function](#)" on page 414 for details. The only difference is the syntax used to implement the function. This is listed below.

Login Function Example

```
<tr>
  <td>
    <%= ecmlogin() %>
  </td>
</tr>
```

DHTML Menus

Ektron CMS400.NET's Menu feature lets users create and maintain a dropdown menu system for your Web site. The Menu feature lets you display menu options that link to content blocks, library assets, external hyperlinks and submenus.

To learn more about menus, please refer to the "Menu Feature" chapter of the **Ektron CMS400.NET** User Manual.

Additional Documentation in Ektron Knowledge Base

This chapter explains how to implement standard menus that ship with **Ektron CMS400.NET**. Ektron's Web site also provides Knowledge Base articles that explain how to extend the standard menu features.

Displaying a Menu on a Web Page

GenerateMenu and DropMenu

IMPORTANT! The menu functions described below are only one example of how to implement menus. You can implement your own menu system using JavaScript.

Use the `GenerateNextGenMenu` and `DropNextGenMenu` custom functions to display a DHTML menu on a Web page. Place `GenerateNextGenMenu` once on the page to provide an infrastructure for `DropNextGenMenu`. Then, place `DropNextGenMenu` as needed to implement individual menus.

Follow these steps to display a menu on a Web template.

1. Place the following code directly after the page's <body> tag.

```
<%= ecmDHTML_GenerateNextGenMenu("menu id number") %>
```

For example:

```
<%= ecmDHTML_GenerateNextGenMenu("1") %>
```

To display more than one menu, separate each id number with a comma. For example:

```
<%= ecmDHTML_GenerateNextGenMenu("1,2,3") %>
```

2. If the user clicks *text* to display the menu, place the following code within the template file where the text would appear.

```
<%= ecmDHTML_DropNextGenMenu menu ID number,"text user clicks to display menu" %>
```

For example:

```
<%= ecmDHTML_DropNextGenMenu 1,"MyMenu1" %>
```

If the user clicks a *graphic* to invoke the menu, use this syntax:

```
<%= ecmDHTML_DropNextGenMenu menu ID number, '<img src=graphic file name>' %>
```

For example:

```
<%= ecmDHTML_DropNextGenMenu 1,'<img src=mymenugraphic.jpg>' %>
```

Customizing the Menu's Appearance

You can customize a menu's appearance, such as the font style, color, and size. So, you have the flexibility to coordinate a menu with the design your Web site.

To customize a menu in **Ektron CMS400.NET**, you can change the ek_Menu.js file.

ek_Menu.js

You can change the look and feel of a menu by changing the ek_menu.js file. If you used the default settings when installing **Ektron CMS400.NET**, this file is in *webroot/cms400demo/Workarea/java/*. For a description of the ek_menu.js file, see ["ek_Menu.js" on page 496](#).

XML Menu Function

The ecmGetMenuXML function wraps XML around the properties of a menu created in the Ektron CMS400.NET Workarea. Once created you can use the XML to display the menu. The ecmGetMenuXML function allows you to easily define how the menu data appears on the Web page.

```
<%= ecmGetMenuXML(int id) %>
```

This function is basically the same as the XML Menu function in ASP. See ["XML Menu" on page 499](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

```
<%= ecmGetMenuXML(1) %>
```

Metadata Function

The metadata function is used to display the metadata for one or more content blocks in the Ektron CMS400.NET Web site. Inserting this custom function allows you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom function looks like:

`<%= ecmMetadata(String itemlist, java.lang.object spare1) %>`

This function is basically the same as the Metadata function in ASP. See ["Metadata Function" on page 488](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP examples are below.

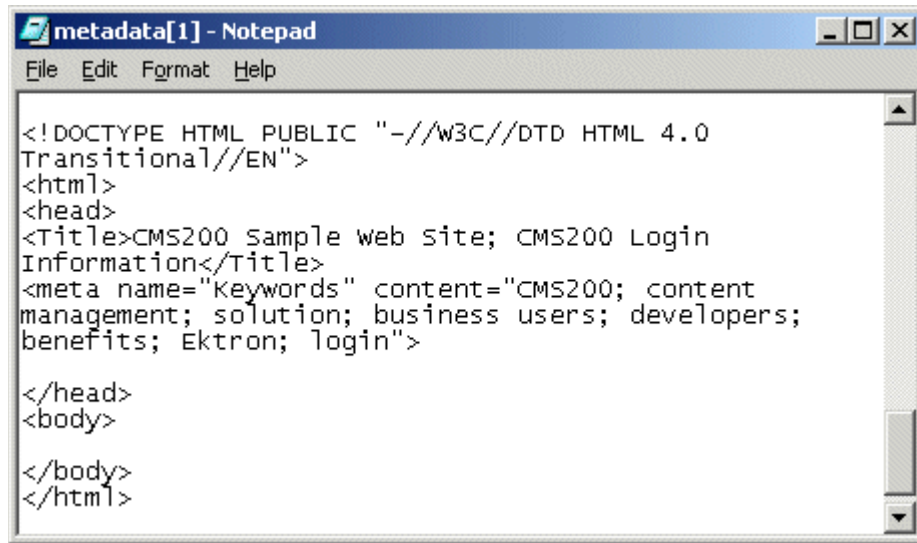
To include both content blocks' metadata in the web page, an ecmMetadata call would look like this.

```
<html>
  <head>
    <%= ecmMetadata ("[1][2;Title,Content-Type]", "") %>
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

In the above example the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). Once the Web page is viewed the Web page source will contain the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

<%= ecmMetaData ("[1][2]", "") %>



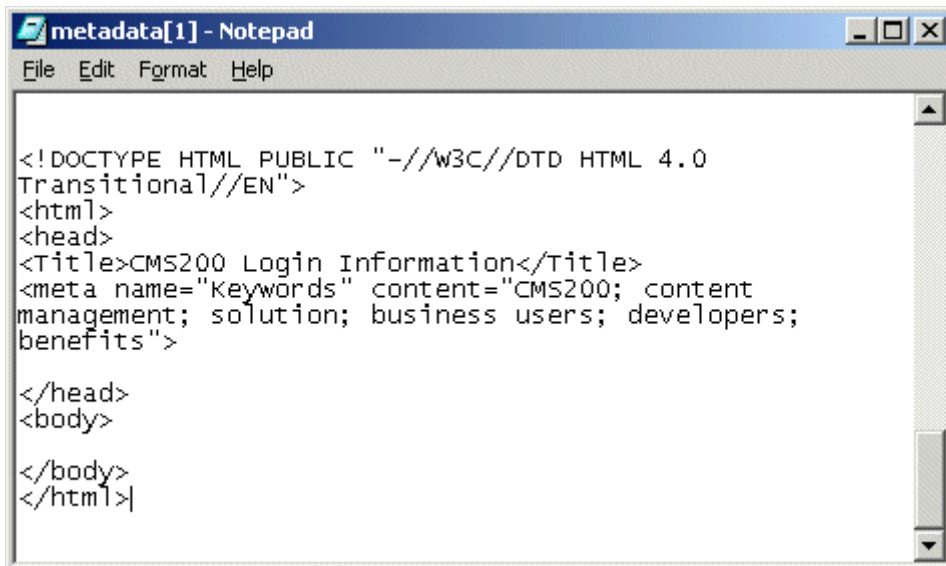
```
metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site; CMS200 Login
Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits; Ektron; login">

</head>
<body>

</body>
</html>
```

<%= ecmMetaData ("[1;title][2;keywords]", "") %>



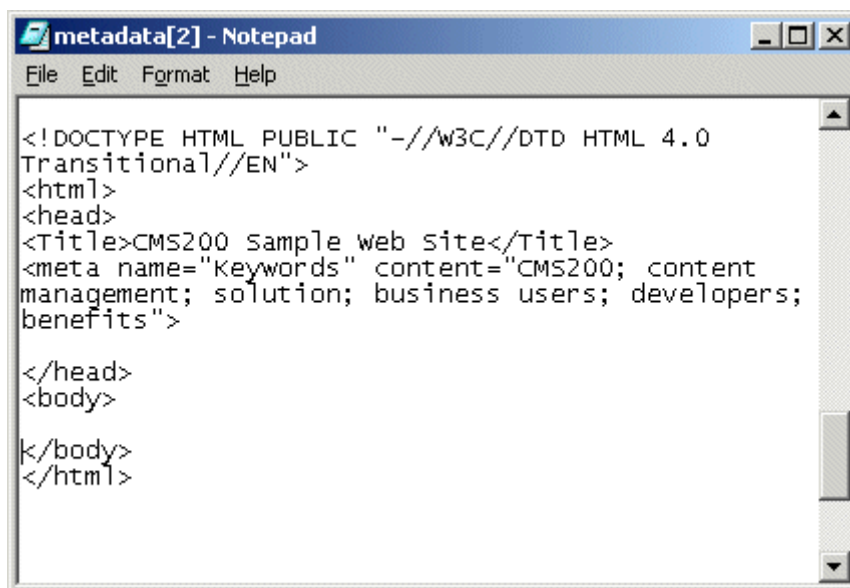
```
metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>
```

```
<%= ecmMetaData ("[1][2;title,keywords]", "") %>
```



Search Function

The search function is used to display a text box with a search button on the template. When seen on the template on the Web page, a user can enter text and click the search button to search through all the content on the Web site.

Ektron has created two search functions:

- ecmSearch
- ecmSearchNoFormTag

Both functions have the same parameters and perform the same search. The only difference is ecmSearch automatically inserts form tags, ecmSearchNoFormTag does not.

The following is an example of form tags.

```
<form name="ecmsearch" method="post" action='search-  
display.asp'>  
  
</form>
```

When using ecmSearch, form tags are added automatically. When using ecmSearchNoFormTag you need to add the tags manually

Here is the format for the search function. You can interchange ecmSearch and ecmSearchNoFormTag when using this function.

```
<%= ecmSearch(java.lang.String startingFolder, boolean recursive, java.lang.String target, int
textBoxSize, int maxCharacters, java.lang.String buttonImageScr, java.lang.String buttonText,
java.lang.String fontFace, java.lang.String fontColor, java.lang.String fontSize, boolean
horizontal, java.lang.Object spare1) %>
```

This function is basically the same as the Search function in ASP. See ["Search Functions" on page 428](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

Search Function Example

```
<TR>
  <TD>
    <%= ecmSearch("\\",true,"search-
display.jsp",25,200,"","Search","Verdana","#808080","2",false,"") %>
  </TD>
</TR>
```

Search Display Function

The search display custom function creates the area where all the search results are shown after the search has been executed.

Here is the format of the ecmSearchDisplay function:

```
<%= ecmSearchDisplay(int maxNumber, String styleInfo, boolean ShowDate,
boolean showSummary) %>
```

This function is basically the same as the Search Display function in ASP. See ["Search Display Function" on page 438](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

Search Display Function Example

```
<tr>
  <td>
    <%= ecmSearchDisplay(0,"",true,true) %>
  </td>
</tr>
```

List Summary Function

The list summary function is used to display a list of content summaries on a Web page. Here is the format of the list summary function.

```
<%= ecmListSummary (String folder, boolean recursive, boolean showSummary,
String styleInfo, String OrderBy, String showInfo, int maxNumber, String objType,
String optionList, String summaryType) %>
```

This function is basically the same as the List Summary function in ASP. See ["List Summary Function" on page 442](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

List Summary Example

```
<TR>
  <TD>
    <%= ecmListSummary ("Marketing\News",false,true,"",
      "Title,asc","DateModified",0,"","") %>
  </TD>
</TR>
```

XML List Summary Function

The XML List Summary function wraps XML around the properties of a list of content, created as a list summary in the Ektron CMS400.NET Workarea. Once created, you can use the XML to display the list summary information. The ecmListSummaryXML function allows you to easily define how the list summary data appears on the Web page.

```
<%= ecmListSummaryXML (String Folder, int Recursive, String OrderBy, int MaxNumber,
String ObjType, String SummaryType) %>
```

This function is basically the same as the List Summary XML function in ASP. See ["XML List Summary Function" on page 450](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

```
<%=
ecmListSummaryXML("CMS400Demo",0,"Title,asc",0,"folder",""); ?>
```

RSS List Summary Function

The RSS summary function is used to create an XML list of content summaries for RSS Syndication. Shown here is the format of the RSS list summary function:

```
<%= ecmRssSummary (java.lang.String Folder, boolean Recursive, java.lang.String OrderBy,
int MaxNumber) %>
```

This function is basically the same as the RSS Summary function in ASP. See ["RSS Summary Function" on page 460](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

```
<%= ecmRssSummary( "\\CMS400Demo\\", false, "Title,asc", 0) %>
```

Single Summary Function

The single summary function is used to display the summary of one content block in the Ektron CMS400.NET Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary function:

```
<%= ecmRssSummary (int contentID, boolean showSummary, java.lang.String showInfo, java.lang.String styleInfo, java.lang.String spare1) %>
```

This function is basically the same as the Single Summary function in ASP. See ["Single Summary Function" on page 456](#) for details.

The only difference between the JSP function and the ASP one is the syntax for implementing it. The JSP example is below.

Single Summary Example

```
<TR>
  <TD>
    <%= ecmSingleSummary ("14",true,"","","") %>
  </TD>
</TR>
```

Web Services

Introduction to Web Services

Web Services (WS) are reusable software components that can be consumed by variety of clients, independent of operating system, programming language, or Web server, using standard Internet technologies. Because Web Services allow heterogeneous systems to communicate with each other in a standard way, they have received quick acceptance for data interchange over the Web.

It's easy for anyone to integrate your WS into their application. The client application can be a Web application, a Windows application, a Java application, a Visual Studio.net application, or a host of others -- it does not matter as long as it supports Web Services.

A Web Service consists of one or more methods that expose functionality. In this sense, a Web Service is an API that can be accessed using SOAP over HTTP. (SOAP is an XML-based, lightweight protocol that defines the message format of Web Services.)

For example, you might develop a purchasing application that automatically obtains price information from several vendors, lets the user select a vendor, submits the order and then tracks the shipment until it is received. The vendor application, in addition to exposing its services on the Web, might in turn use XML Web services to check a customer's credit, charge the customer's account and set up the shipment with a shipping company.

This manual does not explain how Web Services work. To learn more about them, try these sites.

- <http://msdn.microsoft.com/webservices/understanding/readme/default.aspx>
- <http://aspalliance.com/jnuckolls/articles/introws/default.aspx>
- <http://aspnet.4guysfromrolla.com/articles/062602-1.aspx>
- <http://www.15seconds.com/Issue/010430.htm>

CMS400.NET's Implementation of Web Services

CMS400.NET supports XML Web Services. This means that you can create pages that offer your content to any Web-enabled device, and you can create pages that consume existing Web Services. CMS400.NET's Web Services feature can return data separated from presentation. In combination with XML support, this feature lets consumers retrieve your XML data independent of presentation information and act on it however they see fit.

Benefits of Using Web Services

Web Services make it easy to connect with partners. They can also deliver more personal, integrated experiences to users via smart devices, including

PCs. WSs can save time and money by reducing development time, and increase revenue by making your own XML Web services available to others.

WS Components and CMS400.NET Architecture

Web Services require the installation of the .NET Framework on the server.

Ektron's Web Service, ContentWS.asmx, runs within the .NET Framework. Through the Application API class in the Ektron.cms.commonUI.dll, .NET managed code can communicate with asp.net assembly code. ContentWS.asmx utilizes the ektron.com.commonui.dll file to provide access to the Content Business Objects (content blocks, menus, collections, list summaries, etc.).

Software and Hardware Requirements

- .NET framework, version 1.1 SP2, must be installed on the server (for more information, see <http://msdn.microsoft.com/netframework/>).
- Ektron CMS400.NET, version 4.5 or higher

Installation of Files

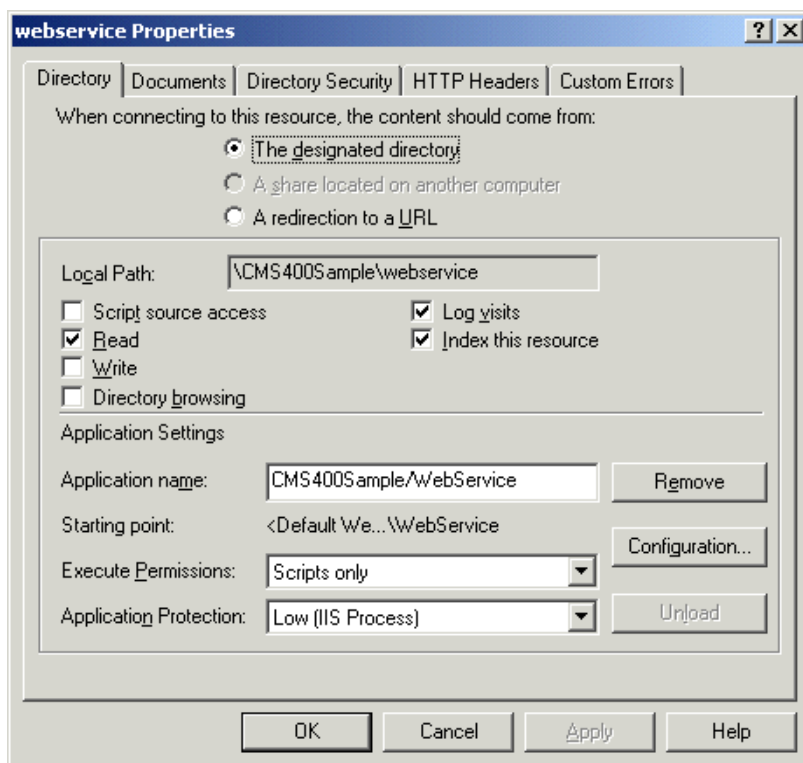
Below is a list of the files and their locations that make up the Web Services feature. They are installed when you install CMS400.NET.

File or folder location	Description
webroot/cms400demo/Workarea/webservices/ContentWS.asmx	Entry point for Web Service. This Web page displays examples of all methods.
webroot/cms400demo/bin	Directory for compiled Web Service code
webroot/cms400demo/bin/WebService.dll	Web Service dll

Setup Instructions

After you install CMS400.NET, you need to open IIS, select cms400demo/workarea, and make it an application. To make a folder an application, follow these steps.

1. Open IIS.
2. Right click one of the folders listed above.
3. Select **Properties**. The Properties dialog appears.



4. Make sure you are on the **Directory** tab, which is selected by default.
5. Press the **Create** button next to the Application Name field.
6. Assign the folder's name in the **Application name** field.
7. Press **OK**.

If you are creating a new directory for your Web site, you must copy the following files into it from `webroot/cms400demo/bin`.

- WebService.dll
- Ektron.Cms.BE.Content.dll
- Ektron.Cms.BE.Font.dll
- Ektron.Cms.BE.Library.dll
- Ektron.Cms.BE.License.dll
- Ektron.Cms.BE.Message.dll
- Ektron.Cms.BE.Module.dll
- Ektron.Cms.BE.Site.dll
- Ektron.Cms.BE.ToolBar.dll
- Ektron.Cms.BE.User.dll

- Ektron.Cms.BE.UserMgmt.dll
- Ektron.Cms.Common.dll
- Ektron.Cms.DataAccess.dll
- Ektron.Cms.DataRW.dll
- Ektron.Cms.Global.dll
- Ektron.Cms.UI.CommonUI.dll
- Ektron.Cms.UI.dll
- Ektron.Cms.WebserviceUI.dll

Providing Web Service Information in Your CMS

CMS400.NET Methods

CMS400.NET implements a single Web Service, ContentWS.asmx, that exposes several methods for retrieving CMS400.NET content. These methods mirror those in the WebService.dll file.

The methods determine the *kind* of content that is made available (for example, a content block, a collection of content blocks, a summary list) along with the *presentation* of that information (for example, an HTML string or an XML data structure representing content block information).

Below is a list of methods that ContentWS exposes with a brief description of functionality and differences from the applicationAPI implementation.

All methods are demonstrated on the sample page *webroot/cms400demo/Workarea/webservices/ContentWS.asmx*. For more information, see ["The CMS400.NET Content Sample Page" on page 567](#).

Method	Exposes this applicationApi method as a web service method	Returns	Changes to standard parameter values
ContentBlock	ecmContentBlock	The content block as an HTML string.	
ContentBlockEx	ecmContentBlockEx	The content block as an XML string. If necessary, performs XSLT transformation on content prior to its return.	
GetContentBlock	ecmContentBlock	An XML data structure representing information in the ContentBlock.	

Collection	ecmCollection	A Collection presented as an HTML string whose format is determined by the display function that you specify. You can create a custom display function in a separate .NET dll.	
GetCollection	ecmCollection	An XML data structure representing information about a Collection.	additional parameter: GetHTML. Possible values: 1 -returns html content for each collection item 0 - does not return it
ListSummary	ecmListSummary	List Summary information as an HTML string.	
GetListSummary	ecmListSummary	An XML data structure representing information about the List Summary.	parameters ShowSummary, StyleInfo, ShowInfo not available.
SingleSummary	ecmSingleSummary	Single Summary information as an HTML string.	
GetSingleSummary	ecmSingleSummary	An XML data structure representing the Single Summary information.	parameters StyleInfo, ShowSummary, ShowInfo not available.
MetaData	ecmMetaData	Meta Data information as an HTML string	does not have the unused spare parameter.
GetMetaData	ecmMetaData	An XML data structure representing Meta Data information	does not have the unused spare parameter.
SearchDisplay	ecmSearchDisplay <i>See Also: "The Search Display Method Parameters" on page 570</i>	Search Display information as an HTML string	

GetSearchDisplay	ecmSearchDisplay <i>See Also: "The Search Display Method Parameters" on page 570</i>	An XML data structure representing the Search Display information	StyleInfo and ShowDate parameters not available
GenerateHTMLMenu	ecmGenerateMenu	Menu details as an html string	Parameters as menu ids (comma separated value)
DropHTMLMenu	ecmDropMenu	Required menu as an html string	Parameters: <ul style="list-style-type: none"> • Root menu id (that is, the generateHTMLMenu value) • MenuId - required menuid • title - required title
GetFormBlock	ecmFormBlock	Form information as an HTML string	Parameters: <ul style="list-style-type: none"> • formId • formdata generated through webserviceUI • formtag required flag (set to true to embed <form> tag; otherwise, set to false)
GetEventsByCalendar	ecmEvtCalendar	Calendar information as an HTML string	<ul style="list-style-type: none"> • Calendar id • event id • month • year • url • querystring
Show RandomSummary	ecmShowRandomSummary	Randomly displays the summary content of a collection item.	<ul style="list-style-type: none"> • collection ID

ShowRandomContent	ecmShowRandomContent	Randomly displays the content of a content block in a specified collection	<ul style="list-style-type: none"> • collection ID • xsItID
AddContent	ecmAddContent	Lets you programmatically add a new content block to Ektron CMS400 without logging in.	<ul style="list-style-type: none"> • Username • password • Domain • content_title • content_comment • ContentHTML • SummaryHTML • ContentLanguage • FolderID • GoLive • EndDate • MetaInfo • ErrString

The CMS400.NET Content Sample Page

The CMS400.NET content sample page (`webroot\cms400demo\Workarea\webservices\ContentWS.asmx`) lists the methods described in the table above. You can access it by signing on to the developer sample site <http://localhost/cms400demo/developer/default.aspx>) and clicking Web Services from the bottom of the left frame (illustrated below).



ektron CMS400.net

Welcome to the Ektron CMS400.NET Developer Section.

Ektron has compiled a list of several different examples of developing your Web site with Ektron CMS400.NET.

The individual examples, which can be accessed by clicking on their title in the left navigation menu, show how easy it is to create robust Web pages by using "out-of-the-box" functionality including server controls. Other examples show how easy it is to programatically produce Web applications by building on these server controls.

Click each method on the page to see sample calls and output using various protocols.

When you open the detail page, you see **Test** followed by parameters.

ecmGetContentBlock

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
ContentId:	<input type="text"/>
UserID:	<input type="text"/>
SitePath:	<input type="text"/>
Preview:	<input type="text"/>
SiteLanguage:	<input type="text"/>
<input type="button" value="Invoke"/>	

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /CMS400Example/Workarea/Webservices/ContentWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.ektron.com/cms/WebService/ContentWS/ecmGetContentBlock"
```

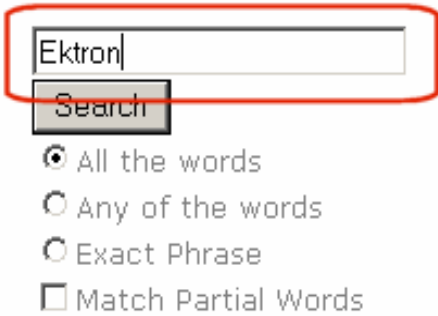
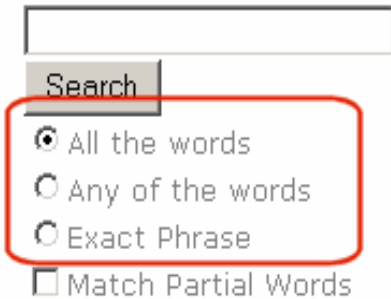
The top section of the page lets you test the method by using the HTTP Get protocol to retrieve data from the sample database. To do this, you must insert valid parameter values.

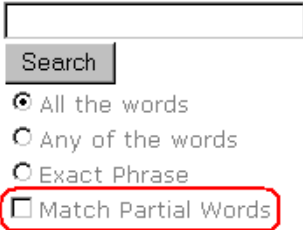
The next section of the CMS400.NET content sample page displays the following kinds of sample output.

- A sample SOAP request and response.
- A sample HTTP GET request and response.
- A sample HTTP POST request and response.

Review the test pages to learn the kind of content you can retrieve from this method.

The Search Display Method Parameters

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Search Text	n/a	<p>Search text used in search method.</p> 
Search Type	n/a	<p>Type of search used in search method.</p>  <p>Options and - all the words or - any of the words exact phrase - exact phrase</p> <p><u>Note: This parameter is handled automatically by CMS when displayed on Web page. For Web Services, however, this needs to be defined.</u></p>
Recursive	Search	<p>Defines whether search is recursive .</p> <p>1 - recursive search 0 - non-recursive search</p>

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Starting Folder	Search	<p>Describes path to folder in which search begins. · Here are some examples.</p> <ul style="list-style-type: none"> • \ - root content folder • \Products\RC Cars - begins search in the RC Cars folder.
Allow Fragments	n/a	<p>Selected by user on page that performs search.</p>  <p>Options</p> <p>0 (false) - do not allow fragments.</p> <p>1 (true) - allow fragments.</p> <p>If fragments are allowed, and the user enters "AT" as the search string, the search returns topics with these words: pat, ate, hatter.</p> <p>If fragments are <i>not</i> allowed, and the user enters "AT" as the search string, the search only returns topics with the word: at.</p> <hr/> <p>Note: This parameter is handled automatically by CMS when displayed on Web page. For Web Services, this needs to be defined.</p> <hr/>
Max Number	Search Display	<p>Specify the maximum number of topics to display when a search is performed. For example:</p> <p>0 - unlimited</p> <p>1 - one result</p> <p>7 - seven results</p>

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Style Info	Search Display	Specify an inline style that is applied to the hyperlinked portion of topics that result from a search. For example: "font-family:arial; font-weight:bold; backgroundcolor:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"
Show Date	Search Display	Specify whether the last modified date is displayed with search results. 0 - date is not displayed. 1 - date is displayed.

Sample Output from One Method

The following is an example of the content that can be retrieved from one of the methods, `GetContentBlock`. As you can see, the HTML content is provided within the `<ContentHtml>` XML element.

GetContentBlock

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentBlockResult xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:ektron:webservices:content:cms400:v45">
  <ErrorMessage />
  <Item>
    <ContentTitle>Plastic Molder #123</ContentTitle>
    <ContentHtml>
      <?xml version="1.0"?> <jobposting> <job-title>Plastics Molder</job-title> <job-number>123</
job-number> <description><p>RC International, a leading supplier of consumer level radio
controlled vehicles, is looking for an experienced plastics molder to join our team. The
candidate will join our highly skilled team of enthusiastic molders to bring to life our next
generation products. RC International is experiencing extremely rapid growth and offers many
opportunities for advancement.</p></description> <qualifications> <ul> <li>3 years experience
in commercial grade plastic molding technologies.</li> <li>Excellent written and oral
communication skills.</li> <li>Ability to work with very little supervision.</li> <li>An AS
degree in chemistry.</li> </ul> </qualifications> <offering> <p>First and second shift
opportunities exist. RC International offers competitive pay, an extensive benefits package
including a 401k plan, profit sharing, and health coverage.</p> </offering> <contact>
<contact-name>Human Resources Manager</contact-name> <contact-email>jobs@CMS400site.com</
contact-email> <contact-fax>1-555-555-5555</contact-fax> </contact> </jobposting>
</ContentHtml>
    <Xslt1>C:\Inetpub\wwwroot\cms400demo\XMLFiles\xslt\rc_jobs_display.xsl</Xslt1>
    <Xslt2 />
    <Xslt3 />
    <Xslt4 />
    <Xslt5 />
    <DefaultXslt>1</DefaultXslt>
  </Item>
</ContentBlockResult>
```

The News.aspx Sample (non-XML)

To help you understand how the samples work, this section explains the news sample .NET page, news.aspx, which is installed to *webroot/cms400demo/* WebServiceDemo/. Specifically, four key sections of code are explained:

- [Identifying the Code Behind File](#)
- [Setting Up an Instance of Web Services](#)
- [Retrieving the Web Service's URL](#)
- [Getting Content Blocks](#)

Identifying the Code Behind File

Here is the first line in the news.aspx file:

```
<%@ Page CodeBehind="news.aspx.vb" Language="vb" AutoEventWireup="false"
Inherits="WebServicesDemo.news" %>
```

This line indicates that the page's functionality is set up in the Code Behind Visual Basic file, news.aspx.vb. The rest of the news.aspx file contains only page display information.

Setting Up an Instance of Web Services

The following code in news.aspx.vb sets up an instance of the Web Services. Every page that uses Web Services must contain such a declaration.

```
Dim wsCms400WebService As New ContentWS.ContentWS
```

Getting Content Blocks

The following lines of news.aspx.vb code retrieve all content blocks in the sample database's news folder formatted as an HTML string.

```
If (iId > 0) Then
    strHTML = wsCms400WebService.ContentBlock(iId)
    strHTML = Replace(strHTML, ".asp", ".aspx")
    NewsInfo.InnerHtml = strHTML

Else
    strHTML = wsCms400WebService.ListSummary("\Marketing\News", 0, 1, "", "Title,asc",
    "DateModified", 0, "", "")
    strHTML = Replace(strHTML, "news.asp", "WebServiceDemo/news.aspx")
    NewsInfo.InnerHtml = strHTML
End If
```

NOTE Since the database was created with .NET, you could use this code to get the HTML for the requested ID:

```
NewsInfo.InnerHtml = wsCms400WebService.ContentBlock(iId)
```

As you can see, we are assigning to the variable NewsInfo.InnerHtml the Web service (wsCms400WebService) followed by the ContentBlock method. (For a list of methods, see ["CMS400.NET Methods" on page 564.](#))

If the user selects a content block on the List Summary page (that is, clicks the blue text of the content title), that action passes an id value. The blue text is circled in the illustration below.

RC International Ships RC Sportster (8/27/2002 2:46:12 PM)

RC International, an emerging leader in the RC racing industry, has begun shipments of its Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer

A new page appears, using the ContentBlock method to display the selected block.

```
If (iId > 0) Then  
    strHTML = wsCms400WebService.ContentBlock(iId)
```

The ContentBlock method retrieves the content block as an HTML string, so the result looks like this.

Click [here](#) for a complete list of operations.

GetContentBlock

Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Value
ContentId:	<input type="text"/>
<input type="button" value="Invoke"/>	

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /CMS400Example/Workarea/Webservices/ContentWS.asmx HTTP/1.1  
Host: 192.168.0.219  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
SOAPAction: "urn:ektron:webservices:content:cms300:v21/GetContentBlock"
```

If the user does not select a content block (which is true when the page first appears), the page uses the ListSummary method to display all blocks in the marketing/news folder using the List Summary format.

```
Else  
    strHTML = wsCms400WebService.ListSummary("\Marketing\News", 0, 1, "",  
"Title,asc", "DateModified", 0, "", "")
```

The ListSummary method retrieves the content blocks as an HTML string, so the page looks like this.

Latest News

[RC International Announces the RC Cheetah \(8/27/2002 2:43:01 PM\)](#)

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

[RC International Hosts RC Conference \(8/27/2002 2:44:44 PM\)](#)

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

[RC International Ships RC Sportster \(8/27/2002 2:46:12 PM\)](#)

RC International, an emerging leader in the RC racing industry, has begun shipments of its first product, RC Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer currently on the market.

The News.aspx Sample (XML)

The news.aspx sample file in the /cms400demo/WebServiceDemo folder includes all elements described in ["The News.aspx Sample \(non-XML\)" on page 573](#) plus demonstrations of how to retrieve XML content.

Setting Up Instances of Web Services

In addition to creating an instance of the Web Service, the XML version of news.aspx.vb creates instances of the ContentBlock and ListSummary methods, which are used on the page.

```
Dim wsCms400WebService As New localhost.CMS400Content()  
Dim gcbCms400ContentXml As localhost.ContentBlockResult  
Dim glsCms400ListSummaryResult As localhost.ListSummaryResult
```

Getting Content Blocks

Look at this code from news.aspx.vb.

```
glsCms400ListSummaryResult = wsCms400WebService.GetListSummary("\Marketing\News",  
0, 1, "", "Title,asc", "DateModified", 0, "", "")  
If (Len(glsCms400ListSummaryResult.ErrorString) = 0) Then  
strHTML = "<table><tr><td><b>Content Block Information Return from the      GetListSummary Web  
service call</b></td></tr>"  
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"  
For iLoop = 0 To (glsCms400ListSummaryResult.Item.Length - 1)  
strHTML = strHTML & "<tr><td><b>Content ID: </b>" &  
glsCms400ListSummaryResult.Item(iLoop).ContentID & "</td></tr>"  
strHTML = strHTML & "<tr><td><b>Content Title: </b>" &  
glsCms400ListSummaryResult.Item(iLoop).ContentTitle & "</td></tr>"  
strHTML = strHTML & "<tr><td><b>Date Created: </b>" &  
glsCms400ListSummaryResult.Item(iLoop).DisplayDateCreated & "</td></tr>"  
strHTML = strHTML & "<tr><td><b>Date Modified: </b>" &  
glsCms400ListSummaryResult.Item(iLoop).DisplayDateModified & "</td></tr>"  
strHTML = strHTML & "<tr><td><b>Modified By: </b>" &  
glsCms400ListSummaryResult.Item(iLoop).EditorFname & " " &  
glsCms400ListSummaryResult.Item(iLoop).EditorLname & "</td></tr>"  
strHTML = strHTML & "<tr><td><b>Content Teaser: </b></td></tr>"  
strHTML = strHTML & "<tr><td>" & glsCms400ListSummaryResult.Item(iLoop).ContentTeaser &  
"</td></tr>"
```

```
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"
Next
```

This code loops through all content blocks in the Marketing\News folder. For each block, the page uses the Web Service method `glsCms400ListSummaryResult.Item(iLoop)` to display the following XML data.

- Content ID
- Content Title
- Date Created
- DateModified
- Modified By
- Content Teaser

The result looks like this.

Latest News

Content Block Information Return from the GetListSummary Web service call

Content ID: 11

Content Title: RC International Announces the RC Cheetah

Date Created: 25-Apr-2002 04:43:42 PM

Date Modified: 27-Aug-2002 02:43:01 PM

Modified By: Application Administrator

Content Teaser:

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

Content ID: 10

Content Title: RC International Hosts RC Conference

Date Created: 25-Apr-2002 04:42:55 PM

Date Modified: 27-Aug-2002 02:44:44 PM

Modified By: Application Administrator

Content Teaser:

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

Content ID: 9

Content Title: RC International Ships RC Sportster

Date Created: 25-Apr-2002 04:42:10 PM

Date Modified: 27-Aug-2002 02:46:12 PM

The lower portion of the page also displays the content block in a ListSummary format, as described in ["Getting Content Blocks" on page 573](#).

Latest News

RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM)

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

RC International Hosts RC Conference (8/27/2002 2:44:44 PM)

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

RC International Ships RC Sportster (8/27/2002 2:46:12 PM)

RC International, an emerging leader in the RC racing industry, has begun shipments of its first product, RC Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer currently on the market.

If the user selects a content block and clicks the blue text of any title, a new page appears with

- information about the block (title and any XSLTs used for rendering).
- a demonstration of the GetContentBlock method to display the content block in "raw" HTML or XML format.
- a demonstration of the ContentBlock method to display the content block in processed HTML or XML format.

Here is the code that displays the top section of the page.

```
strHTML = "<table><tr><td><b>Content Block Information from GetContentBlock Web Service: </b></td></tr>"
strHTML = strHTML & "<tr><td><b>Content Block Title: </b>" &
gcbCms400ContentXml.Item.ContentTitle & "</td></tr>"
strHTML = strHTML & "<tr><td><b>Xslt Used for Rendering: </b>"
Select Case (gcbCms400ContentXml.Item.DefaultXslt)
    Case 1
        strTmp = gcbCms400ContentXml.Item.Xslt1
    Case 2
        strTmp = gcbCms400ContentXml.Item.Xslt2
    Case 3
        strTmp = gcbCms400ContentXml.Item.Xslt3
    Case 4
        strTmp = gcbCms400ContentXml.Item.Xslt4
    Case 5
        strTmp = gcbCms400ContentXml.Item.Xslt5
End Select

If (Len(strTmp) = 0) Then
    strTmp = "None Used (HTML Content assumed)"
```

This code uses the gcbCms400ContentXml Web Service to get the content title: gcbCms400ContentXml.Item.ContentTitle.

The code also uses the gcbCms3400ContentXml Web Service to retrieve any XSLTs assigned to the content. If no XSLTs are assigned (as is the case with this sample page), the message "None Used (HTML Content assumed)" appears.

This code results in the following section of the page.

Latest News

Content Block Information from GetContentBlock Web Service:

Content Block Title: RC International Announces the RC Cheetah

Xslt Used for Rendering: None Used (HTML Content assumed)

Next, the page uses the GetContentBlock method to display the content block in “raw” HTML or XML format. Here is the code that displays that section of the page.

```
strHTML = strHTML & "<tr><td><b>Raw HTML (or XML) from GetContentBlock Web Service: </b></td></tr>"
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"
strHTML = strHTML & "<tr><td>" & strRawHtml & "</td></tr>"
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"
```

Finally, the page uses the ContentBlock Web method to display the content block in “processed” format. Here is the code that displays that section of the page.

```
strHTML = strHTML & "<tr><td><b>Processed HTML (or XML) from ContentBlock Web Service: </b></td></tr>"
strHTML = strHTML & "<tr><td>" & strProcessedHtml & "</td></tr>"
strHTML = strHTML & "</table>"
```

Earlier in the news.aspx.vb file, the strProcessedHtml variable was defined as strProcessedHtml = wsCms400WebService.ContentBlock(iId). So, the strProcessedHtml variable uses the wsCms400WebService.ContentBlock method to retrieve content in processed format.

Here is the resulting section of that page. As you can see, the content appears as within a browser.

RC International Announces the RC Cheetah

January 14, 2001 - RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah. After nearly two years of development the RC Cheetah is designed to compete at the highest level in the RC racing arena. The excitement level of the hard core enthusiasts can be felt at all the demonstrations of the RC Cheetah.

"The RC Cheetah price/performance ration is simply incredible", stated Bob Dickerson, RC International VP of engineering. "The RC Cheetah has been devastating its competition in all the pre-release races and we expect that to continue..."

The major industry magazines are agreeing with Mr. Dickerson's assessment. RC Magazine has tested the RC Cheetah head-to-head with the competition's best and they concluded:

"There is no comparison between the RC Cheetah and its nearest competitor. In speed, handling, and price the RC Cheetah out shines them all".

The RC Cheetah is now available at all RC International distributors and major Hobby stores. Please contact RC International to locate a local dealer.

Creating a Page that Provides a Web Service

Below are the typical steps you would follow to create your own page that provides a Web Service.

1. Decide which content blocks you want to provide. Here are some examples:
 - a specific content block (for example, `wsCms400WebService.ContentBlock(2)` would display content block 2 regardless of the URL parameters)
 - all blocks in a folder (the `GetListSummary` method's first parameter is folder)
 - all blocks in a collection (the `Collection` method's first parameter is collection ID)
2. Decide what information you want to provide about those blocks, and how to present it. Review "[CMS400.NET Methods](#)" on page 564 to find the type of content and the kind of information that is available about it.

For example, if you want to display XML data about content (such as content title, date modified, date created), use a method that begins with "Get," such as `GetContentBlock`, `GetCollection`, and `GetListSummary`.
3. Create a new folder within the `cms400demo` folder. This folder and some of its subfolders have files that are needed to run Web Services. Place all of your files within the new folder.
4. Using a development tool that supports Web Service integration (for example, Visual Studio 2005), create a .NET (.aspx) file that manages page layout. The file must refer to a Visual Basic or C# file (via a `CodeBehind` command), which manages the page's functionality.
5. Add to each .NET project a Web Reference to the WSDL file. (by default, `webroot/cms400demo/Workarea/webservices/ContentWS.asmx`). The environment then builds the appropriate proxy class to access the Web Service's methods. You can use the proxy class within the application to invoke those methods. CMS400.NET sample .aspx pages demonstrate how to do this.
6. Provide to your consumers a URL to the .NET (.aspx) file.

Syndicating Web Content as RSS

This section explains how to add RSS syndication to Ektron CMS400.NET for ASP.NET. RSS syndication allows you to prepare Ektron CMS400.NET content for reading by RSS news feed aggregators.

To make this task easier, Ektron provides sample files to illustrate RSS syndication. You should customize the files to meet your needs.

What is RSS Syndication?

RSS stands for Really Simple Syndication, a common method of syndicating Web content as an XML document. There are several versions of RSS. Ektron has implemented version 2.0.

In Ektron CMS400.NET, you use RSS to prepare a collection list or a list summary for reading by an *aggregator*. An aggregator consumes an RSS feed and displays it to the user. (An aggregator is similar in function to a portal.) There are several popular aggregators, including:

- AmphetaDesk (<http://www.disobey.com/amphetadesk/>)
- NewzCrawler (<http://www.newzcrawler.com/>).

The RSS Specification

The RSS specification is a document: <http://backend.userland.com/rss>. Ektron has implemented the specification, but you can customize the RSS classes to support advanced features.

RSS Syndication in Ektron CMS400.NET

Ektron CMS400.NET allows you to syndicate content by using .NET. Syndicating via .NET requires that you have the Microsoft .NET framework installed on your computer.

With Ektron CMS400.NET, you can syndicate a collection and a list summary via RSS.

Syndicating via .NET

Architecture of RSS Syndication

aspx pages

You can use an aspx page as a URL to dynamically create and return RSS documents. To display an RSS feed on a Web site, the site administrator identifies the URL of the aspx page. The aggregator determines how to use the document's elements, such as <title>, <description> and <link>, to display the syndicated content.

The sample aspx templates (RSS_ListSummary.aspx and RSS_Collection.aspx) are installed by default to localhost/cms400demo folder. They use the WebService.ContentWS class to retrieve the collection or list summary data. Then, they create CollectionToRSS20 or ListSummaryToRSS20. These classes fill the RSS 2.0 data classes and serialize them to an XML document, which can be returned as an HTTP response.

Since these pages are part of the CMS400.NET sample, you should modify them or create your own aspx pages for your Web site. Place your custom pages in the yourwebsite/WebServices/ folder so that the web.config file is within the path.

Retrieving a Collection or List Summary as an RSS Feed

The left frame of the corporate.aspx page is hardcoded to retrieve the News Collection. The right frame retrieves a ListSummary.

Creating Your Own aspx Collection Page

To create your own collection page, copy the collection object to an RSS 2.0 object and serialize it. To accomplish this, use these methods.

```
ConvertToRSS(ByVal result CMS400WS.CollectionResult, ByVal
title As String, ByVal description As String) As RSS20.rss

Serialize(Optional ByVal stream As System.IO.Stream = Nothing,
Optional ByVal o As Object = Nothing)
```

Example

```
Dim objRSSConverter As New CollectionToRSS20(Request, Response)
objRSSConverter.ConvertToRSS(objResult,
objResult.CollectionTitle, objResult.CollectionDescription)
objRSSConverter.Serialize()
```

Remarks

The ConvertToRSS method returns the RSS 2.0 object so that you can fill in or modify additional fields as desired before calling Serialize. The Serialize method

has optional parameters that let you send the RSS to a stream other than `HttpResponse`.

You can add code to serialize a different object (such as an RSS 1.0 data class) or to run an XSLT on the serialized XML document before returning it as the HTTP response.

Example

```
Dim objRSSConverter As New CollectionToRSS20(Request, Response)
Dim objRSS20 As rss.RSS20.rss =
objRSSConverter.ConvertToRSS(objResult,
objResult.CollectionTitle, objResult.CollectionDescription)
objRSS20.channel.language = "en-us"
objRSSConverter.Serialize()
```

Creating Your Own aspx List Summary Page

To create your own list summary page, copy a list summary object to an RSS 2.0 object and serialize it. To accomplish this, use these methods.

```
ConvertToRSS(ByVal result As CMS400WS.ListSummaryResult, ByVal
title As String, ByVal description As String) As RSS20.rss

Serialize(Optional ByVal stream As System.IO.Stream = Nothing,
Optional ByVal o As Object = Nothing)
```

Example

```
Dim objRSSConverter As New ListSummaryToRSS20(Request, Response)
objRSSConverter.ConvertToRSS(objResult, "News", "The latest news")
objRSSConverter.Serialize()
```

Remarks

The `ConvertToRSS` method returns the RSS 2.0 object so that you can fill in or modify additional fields as desired before calling `Serialize`. The `Serialize` method has optional parameters that let you send the RSS to a stream other than `HttpResponse`.

You can add code to serialize a different object (such as an RSS 1.0 data class) or to run an XSLT on the serialized XML document prior to returning it as the HTTP response.

Example

```
Dim objRSSConverter As New ListSummaryToRSS20(Request, Response)
Dim objRSS20 As rss.RSS20.rss =
objRSSConverter.ConvertToRSS(objResult, "Title", "Description")
objRSS20.channel.language = "en-us"
objRSSConverter.Serialize()
```

Learning About Visual Studio

This section introduces some background information about using Microsoft's Visual Studio. For more information, use the help feature installed with Visual Studio 2005 and Microsoft's developer Center (<http://msdn.microsoft.com/vstudio/>).

Grid Layout vs. Flow Layout

NOTE The following definitions are from Visual Studio Help:

Grid Layout - Absolute positioning attributes are inserted into elements that are added, and updated in elements that are moved. Elements can be dragged across the Design view surface. The positioning grid and Snap to Grid are available.

Flow Layout - Elements are added without absolute positioning attributes. Web browsers arrange elements in the order that they occur on the page, from top to bottom. You cannot drag elements across the Design view surface or use the positioning grid.

Grid layout is the default, which means that all controls drawn to the Web form in the designer window will have absolute positioning. Here is an example.

```
<body MS_POSITIONING="GridLayout">
  <form id="Form1" method="post" runat="server">
    <asp:Button id="Button1" style="Z-INDEX: 101; LEFT: 160px; POSITION: absolute; TOP: 80px"
runat="server" Text="Button"></asp:Button>
    <asp:Button id="Button2" style="Z-INDEX: 102; LEFT: 480px; POSITION: absolute; TOP: 88px"
runat="server" Text="Button"></asp:Button>
    <asp:GridView id="GridView1" style="Z-INDEX: 103; LEFT: 208px; POSITION: absolute; TOP:
152px" runat="server"></asp:GridView>
  </form>
</body>
```

In Grid layout, you can position your controls like a WYSIWYG editor with no knowledge of HTML. However, because absolute positioning is not rendered consistently by all browsers, the page layout can be flexible based on the size of other controls on the page, and the Web browser window.

When other controls are dynamically populated, such as a GridView, controls that appear beneath it in the Web form would be obscured if they were positioned absolutely at design time. In addition, when utilizing globalization of pages with different languages, the size of text areas can vary and cause obstructions.

Customizing Validation

Customizing Validation Options

In the cms_config.aspx file, you can specify a set of validation options for plain text and calculation field types. (Data in other field types cannot be validated.) For these field types, you can assign standard and custom validation checks. The checks are applied when data is inserted into one of the field types, and when the user tries to save a Data Design document whose fields have validation attributes.

You can modify the standard options and enter your own criteria for each field. You can also establish dependencies between fields. For example, a value is only required for a field if a certain Checkbox field is checked.

Validation Elements in the Configuration Data

The default configuration file includes standard validation options for plain text and calculation field types.

```
<validation name="calculation">
  <choice name="none" treeImg="text">
    <caption localeRef="dlgNV8n" />
    <schema datatype="string" />
  </choice>
</validation>
```

and

```
<validation name="plaintext" visible="false">
  <choice name="none" treeImg="text">
    <caption localeRef="dlgNV8n" />
  </choice>
</validation>
```

The validation sections let you control the drop-down list of validation choices for a field type (for example, Plain Text). Here is an overview of that section of the cms_config.aspx file.

```
<datadesign>
  <validation> (0 or more)
    <choice> (0 or more)
      <caption>
      <schema>
      <calculate>
    </choice>
  </validation> OR
  <regex> OR
  <script> OR
  <xpath>
  <validate>
    <regex> OR
    <script> OR
    <xpath>
  </validate>
  <errorMessage> (optional)
</datadesign>
```

Details are provided below.

Validation Attributes

Attribute	Description	Possible values
visible	Controls whether the Validation field appears on the field's Properties dialog.	true, false
enabled	Controls whether the Validation field is active or “grayed out” on the field's Properties dialog. If set to true , the Validation field is active; if false , it appears but is grayed out.	true, false
name	The field type to which the validation configuration data applies. A separate <validation> element must exist for each field type.	plaintext, calculation

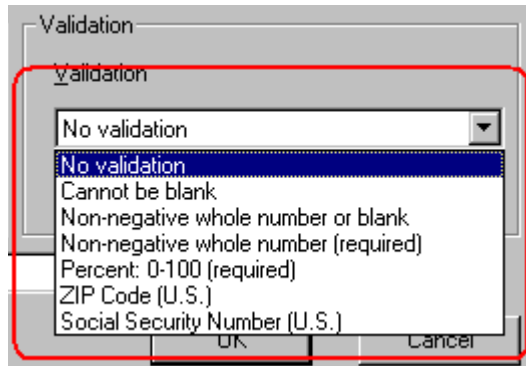
Validation Sub-elements

Element	Description	For more information, see
Choice	The calculation or validation expression for standard fields	"Choice Sub-element" on page 585
Custom	The calculation or validation expression for custom fields	"Defining Custom Validation" on page 590

Choice Sub-element

Every item in the validation drop-down list must be defined within a set of <choice> tags. Within the <choice> tags, you define a caption, a schema, and either a calculation or validation expression.

The default validation choices appear below.



Choice Attributes

Attribute	Description
name	Assign a new name to each choice.
treeImg	<p>The icon to display for this field in the Select Field or Group dialog. See "Icons on the Select Field or Group Screen" on page 595.</p> <p>Unlike toolbar icons, you cannot create your own icons.</p> <p><u>Note: This attribute only applies to the Data Design of an Smart Form -- it does not apply to HTML forms.</u></p>

Choice Sub-Elements

Element	Description	For more information, see
caption	Defines the displayed text for this choice. The attributes and description are the same as <caption> elements for commands.	
schema	Defines a W3C XML schema (WXS) definition for this choice. The definition may be a simple type defined by the datatype attribute and/or a WXS fragment. <u>Note: This attribute only applies to the Data Design of a Smart Form -- it does not apply to HTML forms.</u>	"Defining a Schema Fragment" on page 587
calculate	Defines an expression which normalizes a value prior to checking validation	"Defining a Calculation" on page 588
validate	Defines an expression that determines if a value is valid	"Defining Validation" on page 589

Defining a Schema Fragment

When defining a schema fragment, the "xs" namespace prefix is required for WXS tags. Also, the WXS fragment must be valid for inclusion in an xs:restriction (simple type). That is,

```
<xs:simpleType>
  <xs:restriction>
    ...fragment...
  </xs:restriction>
</xs:simpleType>
```

Simple Data Types

The simple data types are defined by W3C XML Schema definition language 1.0.

NOTE The datatype value should *not* include a namespace prefix. For example datatype="string" is correct; datatype="xs:string" is incorrect

Examples

- Using only a datatype attribute
`<schema datatype="string" />`

- Using only a schema fragment

```
<schema>
  <xs:simpleType>
    <xs:union memberTypes="xs:nonNegativeInteger">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:length value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</schema>
```

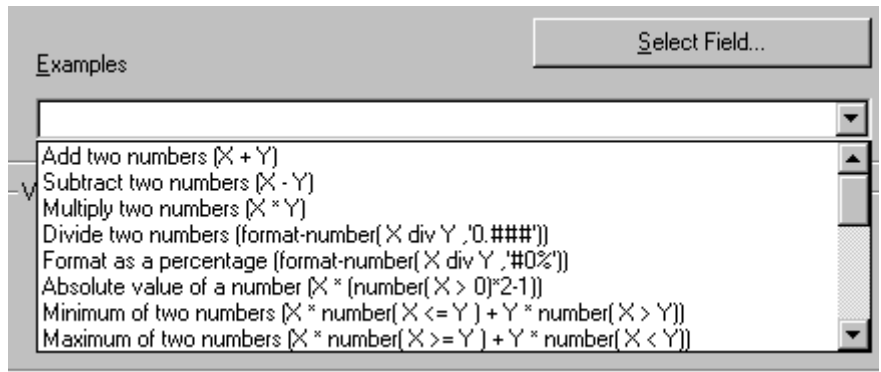
- Using a datatype attribute *and* a schema fragment

```
<schema datatype="string">
  <xs:minLength value="1" />
</schema>
```

Defining a Calculation

Use the `<calculate>` element to define an expression that normalizes a value prior to checking validation. For example, a calculation can truncate digits in a decimal, remove excess white space, or capitalize text. The expression must return a value of the same data type and format as the original value.

Several sample calculations are delivered with **Ektron CMS400.NET**. They appear when the user clicks the **Examples** drop-down on the Calculated Field dialog. These calculations are explained in the “Explanation of Examples” section of the “Using the Data Designer” chapter of the Administrator Manual.



You can only define one expression for `<calculate>` and `<validate>`.

The expression element choices are listed below. Use whichever language is easiest.

- [Regular Expression \(JScript\)](#)
- [JavaScript](#)
- [XPATH](#)

Expression	Description
Regular Expressions (JScript)	
<code><regexp></code>	Defines a regular expression supported by JScript. You can define a regular expression <i>either</i> between the tags <i>or</i> using attributes. If between the tags, the expression must begin with a slash (/) character. The g, i, and m flags are allowed. For example: <code><regexp>/\S+ /</regexp></code> .
<code><regexp patternings-exp-pattern></code>	A regular expression pattern (without the "/" chars).
<code><regexp global="true false"></code>	Specifies whether the pattern matches only the first occurrence or all occurrences within the text. This corresponds to the g flag.
<code><regexp ignorecase="true false"></code>	Specifies if the match is case-sensitive. This corresponds to the "i" flag.
<code><regexp multiline="true false"></code>	Specifies if the match, when using ^ and \$, is applied to each line in text that has multiple lines. This corresponds to the "m" flag.
<code><regexp wholeline="true false"></code>	Specifies whether the pattern applies to the whole text or not. This is the same as placing "^(" at the beginning of the pattern, and ")\$" at the end of the pattern.
JavaScript	
<code><script value="javascript-expression"></code>	Specifies a JavaScript expression to be evaluated. The field's value is available in a property named 'this.text'. For example <code><script value="this.text.toUpperCase()" /></code>
XPATH (See Also: "Learn More about XPath" on page 597)	
<code><xpath select="xpath-expression"></code>	Specifies an XPath expression to be evaluated. The field's value is available using ".".

Defining Validation

Use the `<validate>` element to define an expression that determines if a value is valid. The expression must return a Boolean (true or false) result. For example:

```
<validate>
  <regex>/^\d+$/</regex>
</validate>
```

To construct the `<validate>` element, use the same expression element options as `<calculate>`. See Also: ["Defining a Calculation" on page 588](#)

Defining an Error Message

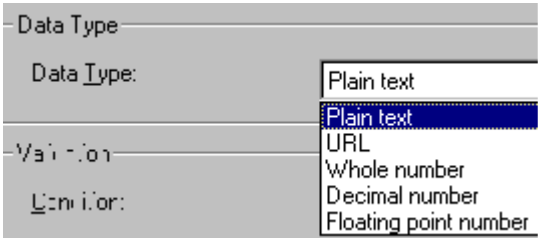
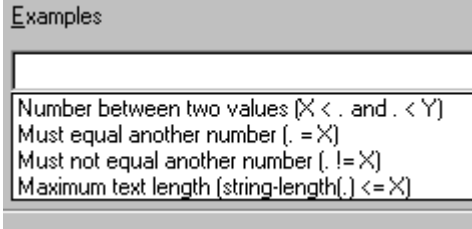
Use the `<errormessage localeRef="id">` element to define a message to display when the data is not valid. For example, "Must be a number between 1 and 10, inclusive."

The text may be within the `<errormessage>` tags or referenced using `localeRef` into the `localeNNNN.xml` file.

Defining Custom Validation

NOTE This section only applies to the Data Design of a Smart Form -- it does not apply to HTML forms.

Use the `<custom>` element to change the standard values that appear in the following fields of the Custom Validation dialog.

Field	Screen image
Data Type	
Examples	

To change the list of options, modify the `<custom>` element of the `cms_config.aspx` file. This section describes the `<custom>` element's attributes and child elements.

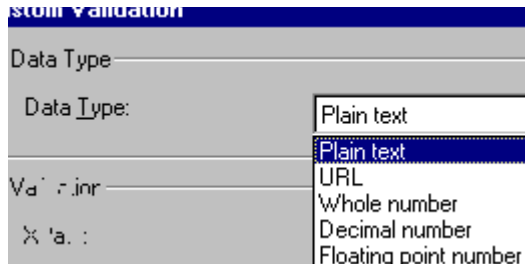
```
<datadesign>
  <validation>
```

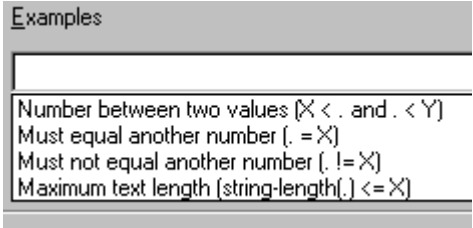
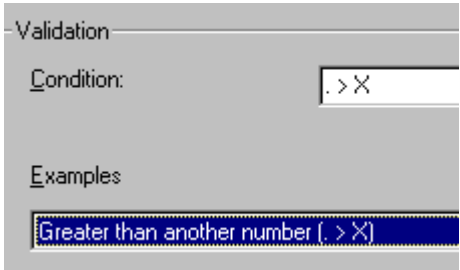
```

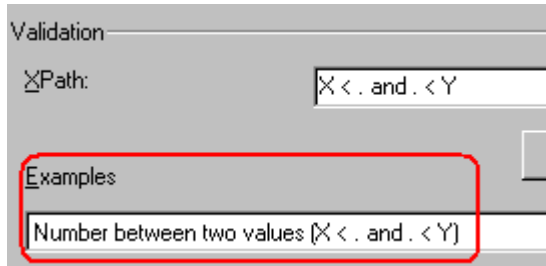
:
<custom> (optional tag)
  <caption>
  <selections name="datatype">
    <listchoice> (0 or more)
  <selections name="examples">
    <listchoice> (0 or more)

```

Element or Attribute	Description
<custom>	Specifies basic data types available when customizing validation. The types are defined in the <listchoice> elements (see below).
<custom visible="true false">	Controls whether the Custom Validation field appears on the Properties dialog.
<custom enabled="true false">	Controls whether the Custom Validation field is active or "grayed out" on the Properties dialog. If set to true , the Custom Validation field is active; if false , it is grayed out.
<caption localeRef="id">	Specifies text to display in the validation drop-down list. The default caption is "(Custom)" .
Selections element for Data Type field	
<selections name="datatype">	The name must be datatype.
<selections enabled="true false">	Controls whether the drop-down list is active or "grayed out" on the Properties dialog.
<selections visible="true false">	Controls whether the drop-down list appears on the Properties dialog.
Listchoice element for Data Type field	
<listchoice>	Defines the values in the Custom Validation dialog Data Type drop-down list.

Element or Attribute	Description
<listchoice value="simple-data-type"	<p>Enter each data type that should appear in the Custom Validation dialog's Data Type drop-down list.</p>  <p>The simple data types are defined by W3C XML Schema definition language 1.0.</p> <p>See Also: "Defining a Schema Fragment" on page 587</p>
<listchoice treelmg="id"	<p>Specifies the icon to display for this field in the Select Field or Group dialog. See "Icons on the Select Field or Group Screen" on page 595.</p> <p>Unlike toolbar icons, you cannot create your own.</p>
<listchoice localeRef="id"	<p>The text that describes this Data Type on the Custom Validation dialog.</p> <p>This element can refer to a string in the localeNNNN.xml file. Or, you can enter the string between the <listchoice> tags.</p>
<listchoice default="true false"	<p>Use this attribute to indicate the default choice in the Custom Validation dialog's Data Type drop-down list.</p>
Selections element for Examples field	
<selections name="examples">	<p>The name must be "examples".</p>

Element or Attribute	Description
<selections enabled="true false"	<p>Determines if the Examples drop-down list and label are active or "grayed out" in the Custom Validation dialog.</p> 
<selections visible="true false"	<p>Determines if the Examples drop-down list and label appear.</p>
Listchoice Element for Examples field	
<listchoice>	<p>Defines the values in the Examples drop-down list.</p>
<listchoice value="xpath-expression-example"	<p>The XPath expression appears in the Examples drop-down list of the Custom Validation dialog.</p> 

Element or Attribute	Description
<listchoice localeRef="id" -	<p>The text that describes the examples on the Custom Validation dialog.</p>  <p>This element can refer to a string in the localeNNNN.xml file. Or, you can enter the string between the <listchoice> tags.</p>

Saving Invalid Documents

Use the `publishinvalid` attribute of the `<standard>` element to determine if an invalid document should be saved. This attribute lets you decide if a user can save an XML document when the data in one or more fields does not satisfy the validation criteria.

In Data Entry mode, an example would be if a field requires a non-negative whole number, but the user does not insert a value in that field.

If the `publishinvalid` attribute's value is `true`, content is not checked for validity when it is saved.

If `false`, the content is checked for validity during a save. If the content is valid, it is saved. If it is invalid, the user is notified, and a custom script can be created to allow the content to be saved or prevent it.

NOTE If the `publishinvalid` attribute is not defined in the `<standard>` element, the default is `true` so that Ektron CMS400.NET is backwards compatible with previous releases.

Custom Script that Handles Saving Invalid Files

To determine whether or not an invalid XML document should be saved, write a client-side script that is called when invalid content is found. The routine should ask if the user wants to save the invalid content. The script should return `True` to save the content, or `False` to abort the save.

Below is an example of such a script.

```
<script language="JavaScript1.2">
<!--
eWebEditPro.instances["MyEditor1"].onerror = myOnErrorHandler;

function myOnErrorHandler()
```

```
{
  if (EWEP_STATUS_INVALID == this.status && "save" == this.event.source)
  {
    var strMsg = "Content is invalid.";
    strMsg += "\nError Code: " + this.event.reason;
    strMsg += "\nError Reason: " + this.event.message;
    alert(strMsg);
    return false; // prevent save
  }
}
//-->
</script>
```

The onerror Event

If the content is invalid, an onerror event is generated. This event provides two additional properties for the event object when it fires:

- reason (a numeric error code)
- message (text describing the error)

As with the regular onerror event, the `source` property is available. You can display the values of these properties in an error message that informs the user why the document is not valid.

Calculated Fields

A Calculated Field lets you include values that are calculated, typically from values in other fields. For example, you can multiply two field values. The equation used is an XPath expression.

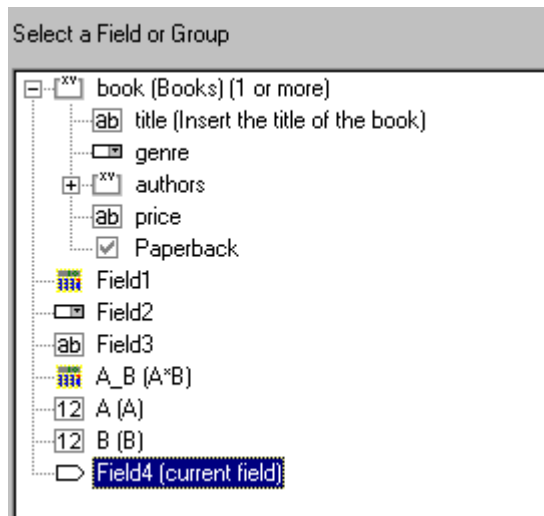
You can validate a Calculated Field, just like the Plain Text Field. For instance, the total of a series of numbers should be less than 100.











For more information about calculated fields, see the **Ektron CMS400.NET** Administrator Manual chapter "Using the Data Designer."




Icons on the Select Field or Group Screen

NOTE [This section only applies to the Data Design of a Smart Form -- it does not apply to HTML forms.](#)

Use the following icons to denote a field on the Select A Field or Group Screen (illustrated below). These are the valid values for the `treeImg` attribute.



Icon	Name
	calculation
	calendar
	checkbox
	droplist
	fieldset
	hidden
	hyperlink
	number
	password
	picture

Icon	Name
	richarea
	text
	textbox

Learn More about XPath

Xpath Operators

Content copied from www.w3schools.com/xpath.

Operator	Description	Example	Result
Numerical expressions perform arithmetic operations on numbers. XPath converts each operand to a number before performing an arithmetic expression.			
+	addition	6 + 4	10
-	subtraction	6 - 4	2
*	multiplication	6 * 4	24
div	division	8 div 4	2
mod	division remainder	5 mod 2	1
Equality/ greater/ less than expressions test equality between two values			
=	equals	price = 9.80	true (if price is 9.80)
!=	is not equal	price! = 9.80	false
<	less than	price < 9.80	false (if price is 9.80)
< =	less than or equal to	price <= 9.80	true

Operator	Description	Example	Result
>	greater than	price > 9.80	false
>=	greater than or equal to	price >= 9.80	true
Boolean expressions compare two values			
or	or	price = 9.80 or price = 9.70	true (if price is 9.80)
and	and	price <=9.80 and price = 9.70	false
Identifying a path and element			
.	the current element	. > 100	true if element exceeds 100
..	the current element's parent	count(../*)	counts the number of elements at the same level as the current element
Grouping and separating			
[]	predicate	../*[0]	../*[0] - returns the parent's first child element
	specify multiple elements	sum(X Y Z)	If X=1 and Y =2 and Z=3, sum(X Y Z) = 6

Xpath Functions

Content copied from www.w3schools.com/xpath.

Operator	Description	Syntax	Example
last	Returns the position number of the last node in the processed node list	number=last()	
position	Returns the position in the node list of the node that is currently being processed	number=position()	

Operator	Description	Syntax	Example
count	Returns the number of nodes in a node-set	number=count(node-set)	
name	Returns the name of a node	string=name(node)	
string	Converts the value argument to a string	string(value)	string(314) Result: '314'
concat	Returns the concatenation of all its arguments	string=concat(val1, val2, ..)	concat('The',' ','XML') Result: 'The XML'
starts-with	Returns true if the first string starts with the second string. Otherwise, it returns false.	bool=starts-with(string,substr)	starts-with('XML','X') Result: true
contains	Returns true if the second string is contained within the first string. Otherwise, it returns false.	bool=contains(val, substr)	contains('XML','X') Result: true
substring-after	Returns the part of the string in the string argument that occurs after the substring in the substr argument	string=substring-after(string,substr)	substring-after('12/10','/') Result: '10'
substring-before	Returns the part of the string in the string argument that occurs before the substring in the substr argument	string=substring-before(string,substr)	substring-before('12/10','/') Result: '12'
substring	Returns a part of the string in the string argument	string=substring(string,start,length)	substring('Beatles',1,4) Result: 'Beat'
string-length	Returns the number of characters in a string	number=string-length(string)	string-length('Beatles') Result: 7
normalize-space	Returns the whitespace-normalized version of a passed string. All leading and trailing whitespace is stripped, and all sequences of whitespace get combined to one single space.	normalize-space('string')	normalize-space(' some text ') would return some text

Operator	Description	Syntax	Example
translate	normalize	string=translate(value,string1,string2)	<ul style="list-style-type: none"> translate('12:30','30','45') Result: '12:45' translate('12:30','03','54') Result: '12:45' translate('12:30','0123','abcd') Result: 'bc:da'
boolean	Converts the value argument to Boolean and returns true or false	bool=boolean(value)	
not	Returns true if the condition argument is false, and false if the condition argument is true	bool=not(condition)	not(false())
true	Returns true	true()	number(true()) Result: 1
false	Returns false	false()	number(false()) Result: 0
lang	Returns true if the language argument matches the language of the xsl:lang element. Otherwise, it returns false.	bool=lang(language)	
number	Converts the value argument to a number	number=number(value)	number('100') Result: 100
sum	Returns the total value of a set of numeric values in a node-set	number=sum(node-set)	sum(/cd/price)
floor	Returns the largest integer that is not greater than the number argument	number=floor(number)	floor(3.14) Result: 3
ceiling	Returns the smallest integer that is not less than the number argument	number=ceiling(number)	ceiling(3.14) Result: 4

Operator	Description	Syntax	Example
round	Rounds the number argument to the nearest integer	integer=round(number)	round(3.14) Result: 3

XPath References

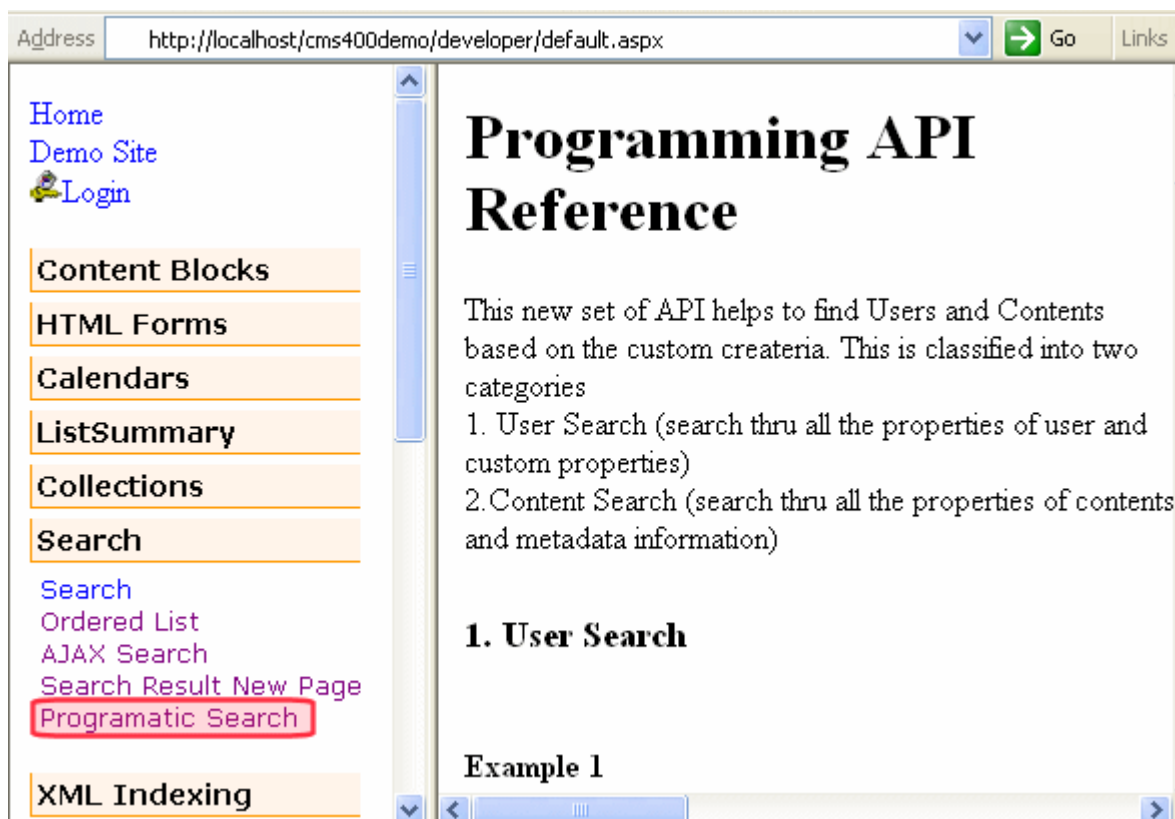
To learn more about XPath, check these Web pages.

Topic	URL
W3C Spec - XPath 1.0 spec	http://www.w3.org/TR/xpath
Tutorial	www.w3schools.com/xpath/default.asp
Quick reference card	www.mulberrytech.com/quickref/XSLTquickref.pdf

Programmatic Search API

Ektron CMS400.NET provides an API that lets you create a programmatic search on users and content. For example, you could use it to find all membership users whose zip code is 03031.

To see an example of the search API, go to the developer's page (<http://your Web server/cms400demo/developer/default.aspx>) and click **Search > Programmatic Search**.



Note that below each example on the page is **run this example**. Click this link to run the search and view the results.

Guidelines for Creating a Search

To learn how to use the Search API, study the examples on the page. Note that examples are provided in both Visual Basic and C#.

Here are guidelines to follow when creating a search.

1. A search consists of one or more conditions. Each condition consists of an operator (and, OR, Like), a value, and the field whose value you are setting. See illustration below.

```
SearchCondition isInEktron = new UserSearchCondition();  
Ektron.setType = SearchType.EQUAL;  
Ektron.setValue = "03031";  
Ektron.setVariable = "customproperties.zip";
```

Condition
Operator
Value
Field

Value can be integer, string, date and bo

WARNING!

If the field is a date, you can only use the following operators: EQUAL, NOT EQUAL, GREATER THAN or LESS THAN.

2. After all conditions are declared, declare the logical relationship between them. In other words, must the search criteria satisfy all or any conditions?

In the sample code below, the search only returns users that satisfy both declared conditions. This is indicated by the AND operator.

```
UserSearchCondition condition = new UserSearchCondition();  
condition.setType = SearchType.AND;  
condition.AddCondition(isInEktron);  
condition.AddCondition(isMemberShip);
```

3. Execute the search.

```
SearchManager search = new SearchManager();  
UserData[] result = search.Execute(condition);
```

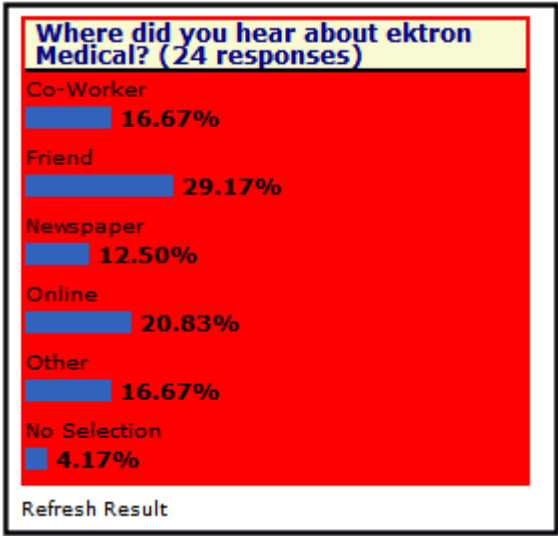
Cascading Style Sheets

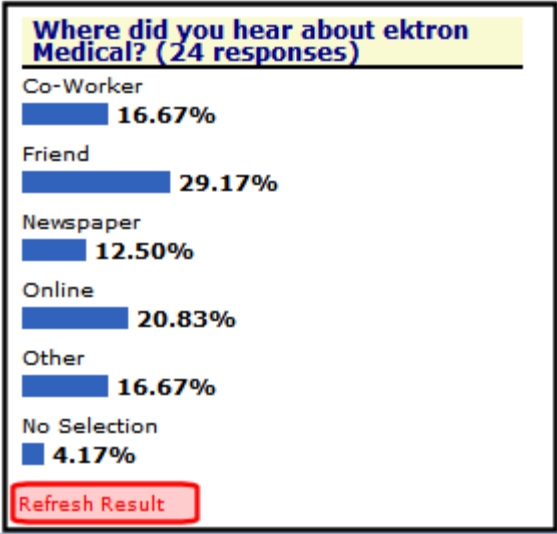
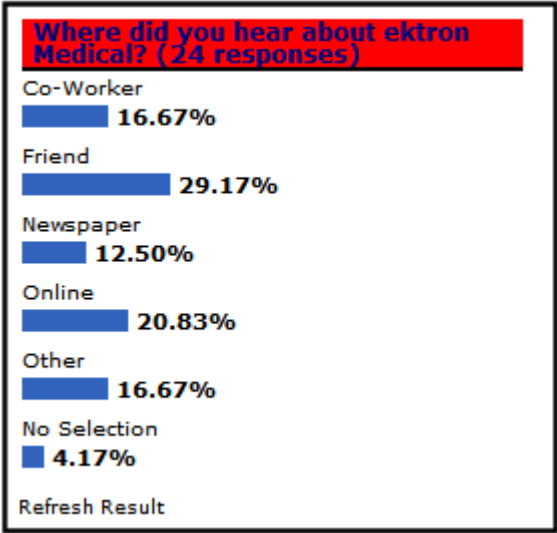
Below are descriptions of the various Cascading Style Sheets used in **Ektron CMS400.NET**.

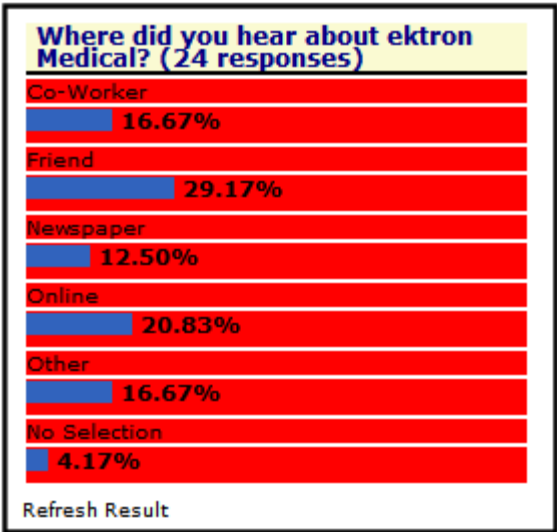
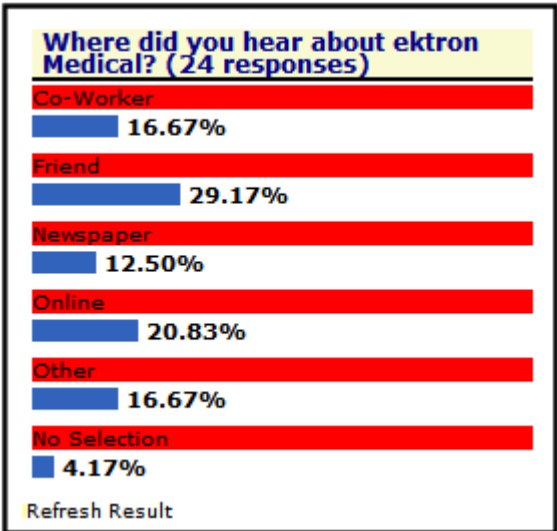
reportchart.css

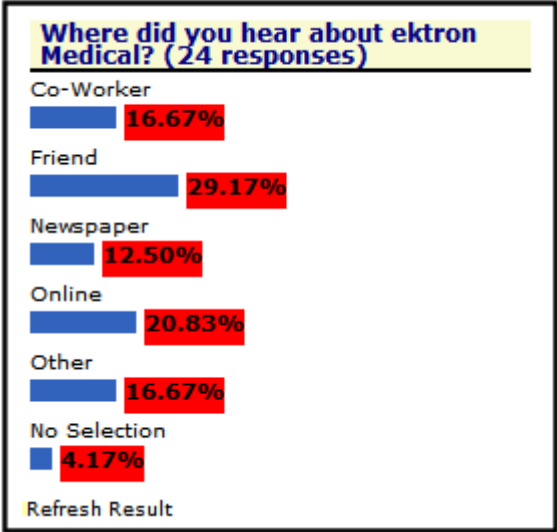

The reportchart.css file allows you to change the look and feel of the postback reports used with forms. This files default location is

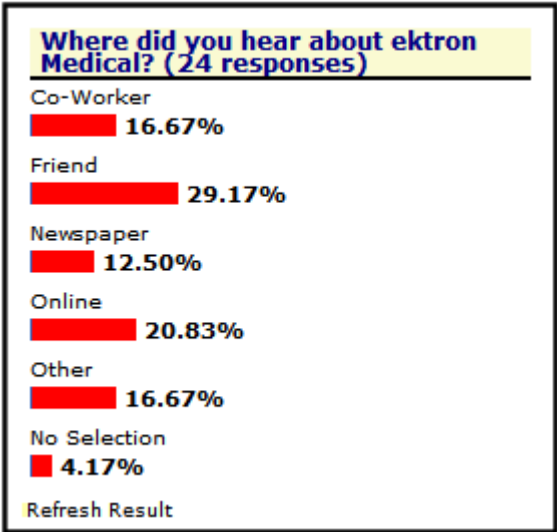
`\webroot\CMS400Demo\Workarea\csslib.`

Class	Controls														
TABLE.tbreport	<p>The area where the report appears. The following example shows the area where the report appears in red.</p>  <table><caption>Where did you hear about ektron Medical? (24 responses)</caption><thead><tr><th>Source</th><th>Percentage</th></tr></thead><tbody><tr><td>Co-Worker</td><td>16.67%</td></tr><tr><td>Friend</td><td>29.17%</td></tr><tr><td>Newspaper</td><td>12.50%</td></tr><tr><td>Online</td><td>20.83%</td></tr><tr><td>Other</td><td>16.67%</td></tr><tr><td>No Selection</td><td>4.17%</td></tr></tbody></table> <p>Refresh Result</p>	Source	Percentage	Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Source	Percentage														
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														

Class	Controls
A.refreshlink	<p>The Refresh Results link that appears below the report. The following example shows the Refresh Results link is highlighted in red.</p> 
A.refreshlink:ACTIVE	The Refresh Results link as you are clicking on it.
A.refreshlink:VISITED	The Refresh Results link after you have clicked on it.
.headreport	<p>The header area where the question and number of responses appear. The following example shows the header area in red.</p> 

Class	Controls														
.tblcol1, .tblcol2	<p>The table columns where the options and results appear. The following example shows the table columns in red.</p>  <table border="1"> <thead> <tr> <th>Where did you hear about ektron Medical? (24 responses)</th> <th></th> </tr> </thead> <tbody> <tr> <td>Co-Worker</td> <td>16.67%</td> </tr> <tr> <td>Friend</td> <td>29.17%</td> </tr> <tr> <td>Newspaper</td> <td>12.50%</td> </tr> <tr> <td>Online</td> <td>20.83%</td> </tr> <tr> <td>Other</td> <td>16.67%</td> </tr> <tr> <td>No Selection</td> <td>4.17%</td> </tr> </tbody> </table> <p>Refresh Result</p>	Where did you hear about ektron Medical? (24 responses)		Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Where did you hear about ektron Medical? (24 responses)															
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														
.optiontextcell	<p>The area where the options appear. The following example shows the option area in red.</p>  <table border="1"> <thead> <tr> <th>Where did you hear about ektron Medical? (24 responses)</th> <th></th> </tr> </thead> <tbody> <tr> <td>Co-Worker</td> <td>16.67%</td> </tr> <tr> <td>Friend</td> <td>29.17%</td> </tr> <tr> <td>Newspaper</td> <td>12.50%</td> </tr> <tr> <td>Online</td> <td>20.83%</td> </tr> <tr> <td>Other</td> <td>16.67%</td> </tr> <tr> <td>No Selection</td> <td>4.17%</td> </tr> </tbody> </table> <p>Refresh Result</p>	Where did you hear about ektron Medical? (24 responses)		Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Where did you hear about ektron Medical? (24 responses)															
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														

Class	Controls														
.percentcell	<p>The area where the result's percentage information appears. The following example shows the result's percentage information area in red.</p>  <p>The screenshot shows a survey titled "Where did you hear about ektron Medical? (24 responses)". It lists seven categories with their respective percentages: Co-Worker (16.67%), Friend (29.17%), Newspaper (12.50%), Online (20.83%), Other (16.67%), No Selection (4.17%), and a Refresh Result button. The percentage values are displayed in red text next to blue bars.</p> <table border="1"> <thead> <tr> <th>Source</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Co-Worker</td> <td>16.67%</td> </tr> <tr> <td>Friend</td> <td>29.17%</td> </tr> <tr> <td>Newspaper</td> <td>12.50%</td> </tr> <tr> <td>Online</td> <td>20.83%</td> </tr> <tr> <td>Other</td> <td>16.67%</td> </tr> <tr> <td>No Selection</td> <td>4.17%</td> </tr> </tbody> </table>	Source	Percentage	Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Source	Percentage														
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														
.barcell	<p>The cell where the bar graph appears. The following example shows the cell area in red.</p>  <p>The screenshot shows the same survey as above, but the entire bar graph area, including the category names, bars, and percentage values, is highlighted in red.</p> <table border="1"> <thead> <tr> <th>Source</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Co-Worker</td> <td>16.67%</td> </tr> <tr> <td>Friend</td> <td>29.17%</td> </tr> <tr> <td>Newspaper</td> <td>12.50%</td> </tr> <tr> <td>Online</td> <td>20.83%</td> </tr> <tr> <td>Other</td> <td>16.67%</td> </tr> <tr> <td>No Selection</td> <td>4.17%</td> </tr> </tbody> </table>	Source	Percentage	Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Source	Percentage														
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														

Class	Controls														
.resultbar	<p>The result's bar graph bar. The following example shows the cell area in red.</p>  <table border="1"> <caption>Where did you hear about ektron Medical? (24 responses)</caption> <thead> <tr> <th>Source</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Co-Worker</td> <td>16.67%</td> </tr> <tr> <td>Friend</td> <td>29.17%</td> </tr> <tr> <td>Newspaper</td> <td>12.50%</td> </tr> <tr> <td>Online</td> <td>20.83%</td> </tr> <tr> <td>Other</td> <td>16.67%</td> </tr> <tr> <td>No Selection</td> <td>4.17%</td> </tr> </tbody> </table>	Source	Percentage	Co-Worker	16.67%	Friend	29.17%	Newspaper	12.50%	Online	20.83%	Other	16.67%	No Selection	4.17%
Source	Percentage														
Co-Worker	16.67%														
Friend	29.17%														
Newspaper	12.50%														
Online	20.83%														
Other	16.67%														
No Selection	4.17%														


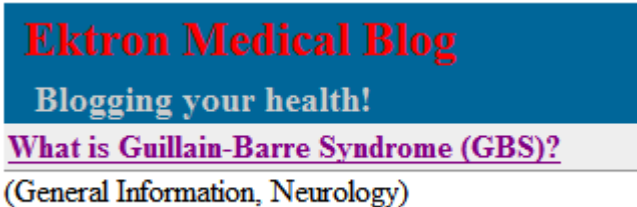

blogs.css

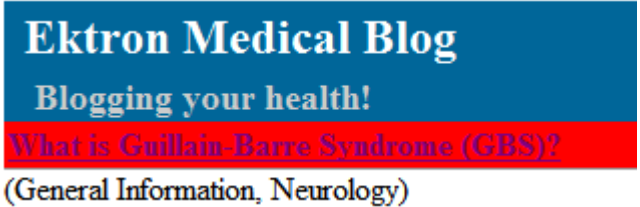

This file is located in `webroot/CMS400Demo/Workarea/csslib`. It is used to customize the look of the Blog server controls.




The table below contains the following sections:




- "Layout Styles" on page 608
- "Header Styles" on page 609
- "Entry Styles" on page 610
- "Calendar Styles" on page 610
- "Blogroll Styles" on page 613
- "Blog Categories Styles" on page 614
- "Blog Recent Posts Styles" on page 615




Style Class	Description
<i>Layout Styles</i>	



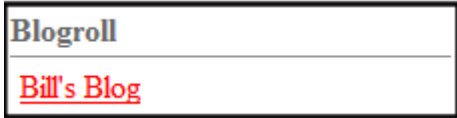

Style Class	Description
div.blog	Affects the blog entry information contained in the Blog server control
div.blogEntries	The border to the right of the blog entries in the Blog server control
div.blogSidebar	The area where the blog calendar, blog roll, categories and recent posts are located in the Blog server control.
Header Styles	
div.blogHeader	<p>The header area for the blog. Affects the Blog and BlogEntries server controls. The following example shows the header area for the blog in red.</p> 
h1.blogTitle	<p>The title of the blog. Affects the Blog and BlogEntries server controls. The following example shows the blog title in red.</p> 
h2.blogSubhead	<p>The subhead (Tagline) of the blog. Affects the Blog and BlogEntries server controls. The following example shows the blog Subhead in red.</p> 


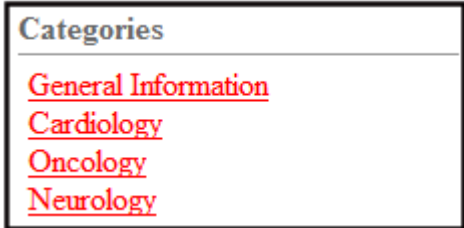

Style Class	Description
span.date	The date in the blog title. Affects the Blog and BlogEntries server controls.
Entry Styles	
div.entry	The border line that separates blog entries. Affects the Blog and BlogEntries server controls.
h3.entryTitle	<p>The title area for blog entries. Affects the Blog and BlogEntries server controls. The following example shows the title area for blog entries in red.</p> 
div.entryBody	The blog entry body. Affects the Blog and BlogEntries server controls.
div.entryFooter	<p>The blog entry footer information. Affects the Blog and BlogEntries server controls. The following example shows the footer information for blog entries in red.</p> 
Calendar Styles	


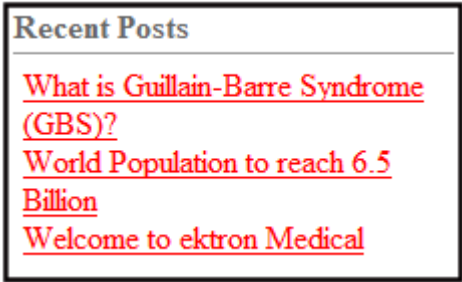
Style Class	Description
table.calendarTable	<p>The table area containing the calendar. Affects the Blog and BlogCalendar server controls. The following example shows the table area in red.</p> 
tr.monthRow	<p>The row that contains month information in the calendar. Affects the Blog and BlogCalendar server controls. The following example shows the month row in red.</p> 
td.prevMonth, td.nextMonth	<p>The area of the calendar for the previous and next month. Affects the Blog and BlogCalendar server controls. The following example shows the previous and next month area in red.</p> 

Style Class	Description
td.prevMonth a, td.nextMonth a	<p>The text information for the previous and next month area of the calendar. Affects the Blog and BlogCalendar server controls. The following example shows the previous and next month text in red.</p>  <p>The image shows a calendar for May 2006. The header bar is blue with the text 'May 2006' in white. On the left and right of the header are red double arrows (≤ and ≥). The days of the week are listed in a row below the header: Sun, Mon, Tue, Wed, Thu, Fri, Sat. The dates 1 through 31 are displayed in a grid below the day names. The dates are blue and underlined. The calendar is set to May 2006, with the 1st of the month on a Monday.</p>
td.currentMonth	<p>The current month area of the calendar. Affects the Blog and BlogCalendar server controls. The following example shows the current month area in red.</p>  <p>The image shows a calendar for May 2006. The header bar is blue with the text 'May 2006' in white. The left and right navigation arrows are also white. The days of the week are listed in a row below the header: Sun, Mon, Tue, Wed, Thu, Fri, Sat. The dates 1 through 31 are displayed in a grid below the day names. The dates are blue and underlined. The calendar is set to May 2006, with the 1st of the month on a Monday.</p>
tr.dayNamesRow	<p>The day names row for the calendar. Affects the Blog and BlogCalendar server controls. The following example shows the day names area in red.</p>  <p>The image shows a calendar for May 2006. The header bar is blue with the text 'May 2006' in white. The left and right navigation arrows are also white. The days of the week are listed in a row below the header: Sun, Mon, Tue, Wed, Thu, Fri, Sat. The dates 1 through 31 are displayed in a grid below the day names. The dates are blue and underlined. The calendar is set to May 2006, with the 1st of the month on a Monday.</p>

Style Class	Description
tr.weekRow td	<p>The rows in the calendar that contain each week's dates. Affects the Blog and BlogCalendar server controls. The following example shows the week row area in red.</p> 
td.inactiveDay	<p>The days with no associated blog posts. Affects the Blog and BlogCalendar server controls. The following example shows the days with no associated blog posts in red.</p> 
td.activeDay	<p>The days with an associated blog post. Affects the Blog and BlogCalendar server controls. The following example shows the days with an associated blog post in red.</p> 
Blogroll Styles	

Style Class	Description
div.blogSidebar h4	<p>The title area for the Blogroll. Affects the Blog and BlogRoll server controls. The following example shows the title area for the Blogroll in red.</p>  <p>Note: Changing this class, changes the title area for Categories and Recent Posts.</p>
div.blogRoll ul	<p>The link area for the Blogroll. Affects the Blog and BlogRoll server controls. The following example shows the link area for the Blogroll in red.</p> 
.blogrolllink	<p>The Blogroll link. Affects the Blog and Blogroll server controls. The following example shows the Blogroll link in red.</p> 
Blog Categories Styles	
div.blogSidebar h4	<p>The title area for the blog's Categories. Affects the Blog and BlogCategories server controls. The following example shows the title area for the blog categories in red.</p>  <p>Note: Changing this class, changes the title area for Blogroll and Recent Posts.</p>

Style Class	Description
div.blogCategories ul	<p>The link area for the blog's Categories. Affects the Blog and BlogCategories server controls. The following example shows the link area for the blog's Categories in red.</p> 
.blogcatlink	<p>The blog's Categories links. Affects the Blog and BlogCategories server controls. The following example shows the blog's Categories links in red.</p> 
Blog Recent Posts Styles	
div.blogSidebar h4	<p>The title area for the blog's Categories. Affects the Blog and BlogRecentPosts server controls. The following example shows the title area for the blog categories in red.</p>  <p>Note: Changing this class, changes the title area for Blogroll and Categories.</p>

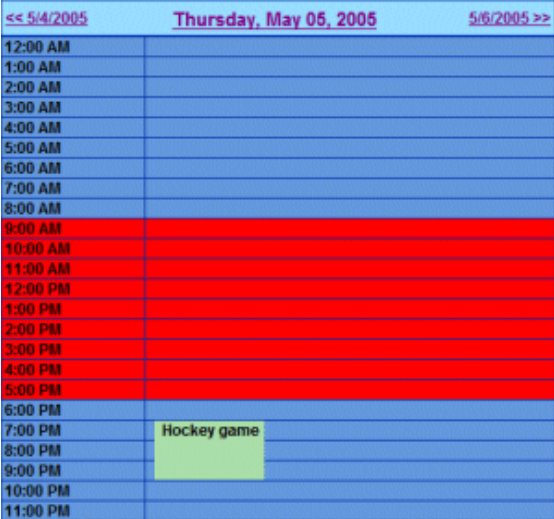

Style Class	Description
div.blogRecentPosts ul	<p>The link area for recent posts. Affects the Blog and BlogRecentPosts server controls. The following example shows the link area for recent posts in red.</p> 
.blogrecentlink	<p>The recent posts links. Affects the Blog and BlogCategories server controls. The following example shows the recent posts links in red.</p> 

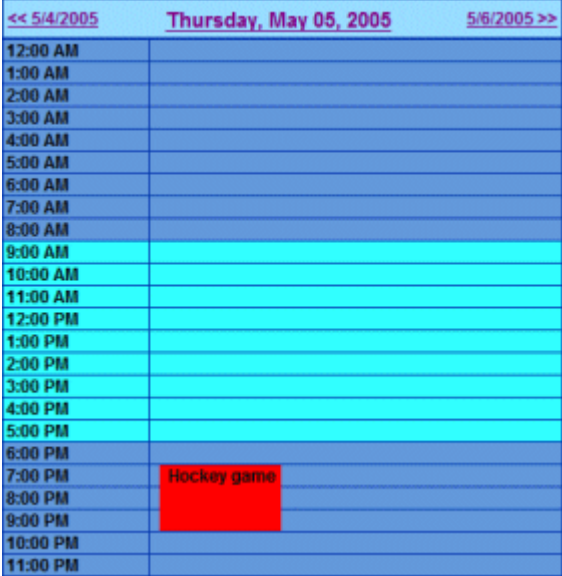
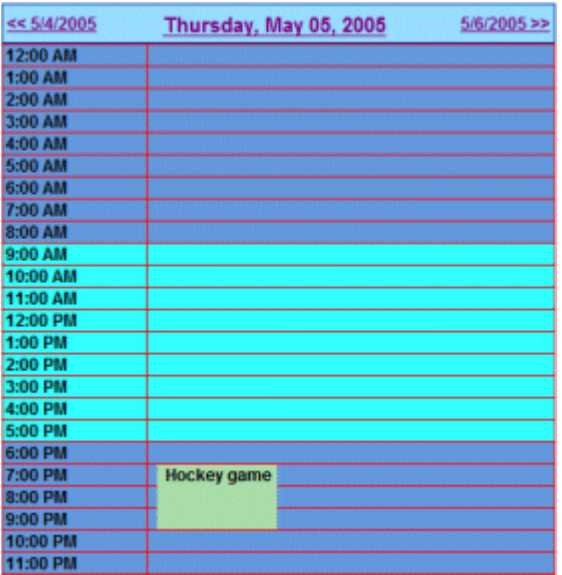
calendarStyles.css


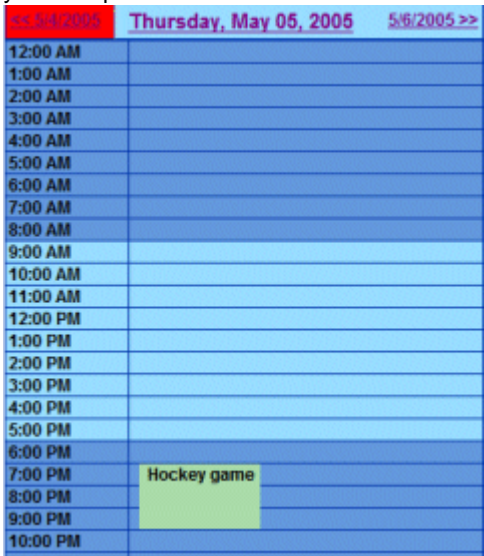
This file is located in `webroot/CMS400Demo/Workarea/csslib`. It is used to customize the look of the Calendar server control.

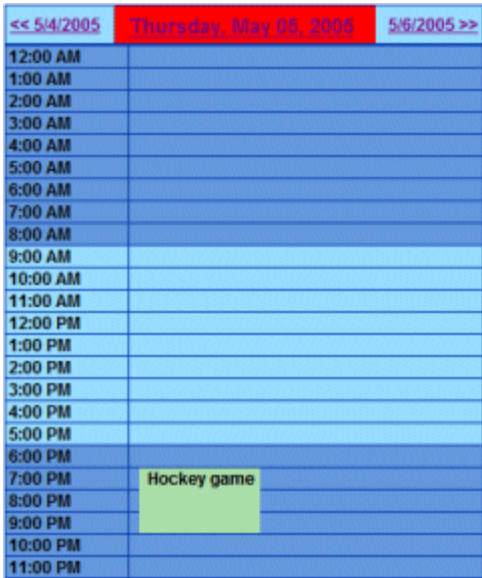
The table below contains the following sections.

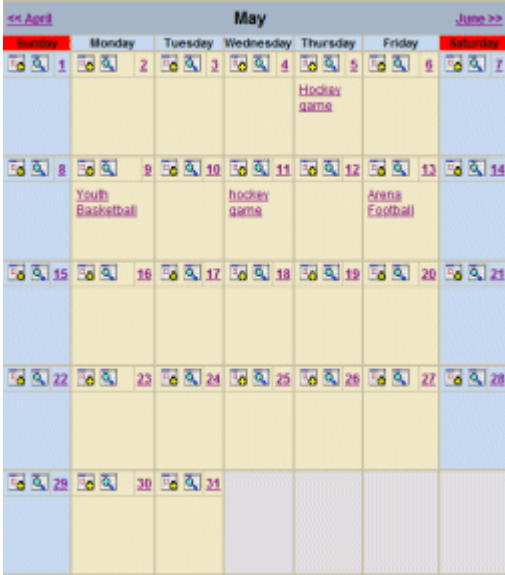
- "Single Day View (Day)" on page 617
- "Month View (month)" on page 620
- "Inline View (inline)" on page 632
- "Month-Up with Event (monthupwithevent)" on page 634
- "Event Type Control (used in month display)" on page 641


Style Class	Description	Attributes and Default Settings
Single Day View (Day)		
.dv_HourCell	<p>The workday hour cells. The following example shows these cells with a red background.</p> 	<ul style="list-style-type: none"> • background: #99DDFF • font-family: Arial, Arial Narrow, Times New Roman • font-size: 9pt • font-weight: bold
.dv_HourCellEve	<p>The evening and morning hour cells. The following example shows these cells with a red background.</p> 	<ul style="list-style-type: none"> • background: #6699DD • font-family: Arial, Arial Narrow, Times New Roman • font-size: 9pt • font-weight: bold

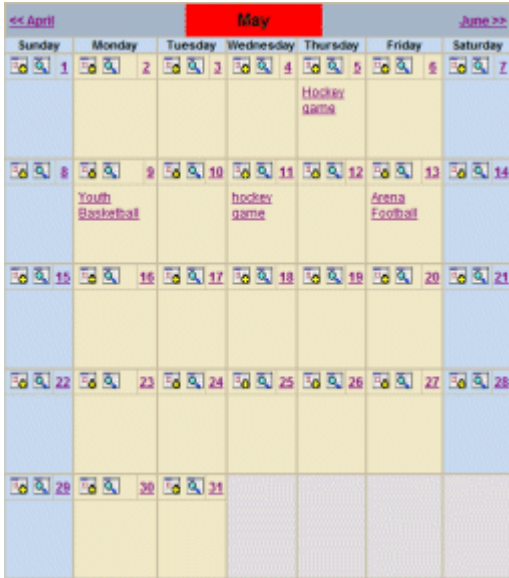

Style Class	Description	Attributes and Default Settings
.dv_EventTitle	<p>An event listed on the daily calendar. The following example shows the event with a red background.</p> 	<ul style="list-style-type: none"> • background: #aaddaa • font-family: Arial, Arial Narrow, Times New Roman • font-size: 9pt • font-weight: bold • cursor: pointer
.dv_BorderCell	<p>The borders of the hour cells. The following example shows the borders in red.</p> 	<ul style="list-style-type: none"> • background: #0033AA

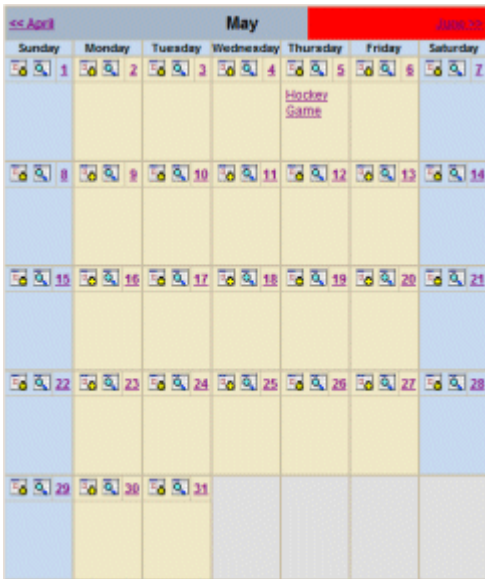
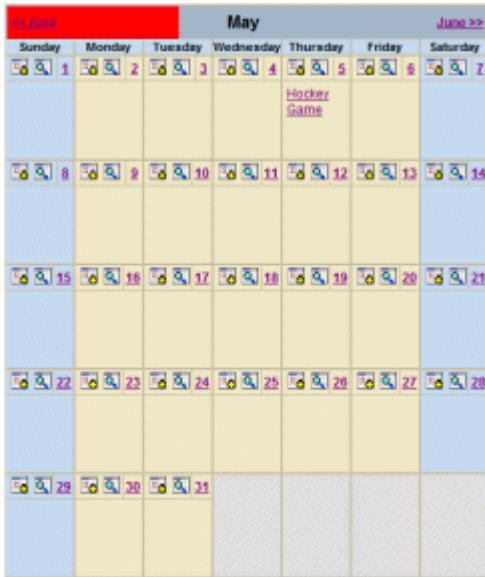
Style Class	Description	Attributes and Default Settings
.dv_DayHeaderNext	<p>The button for moving the calendar to the next day. The following example shows the day header next in red.</p> 	<ul style="list-style-type: none"> • background: #99DDFF • color: #0033AA • font-family: Arial • font-size: 9pt • font-weight: bold • text-align: right
.dv_DayHeaderPrev	<p>The button for moving the calendar to the previous day. The following example shows the day header previous in red.</p> 	<ul style="list-style-type: none"> • background: #99DDFF • color: #0033AA • font-family: Arial • font-size: 9pt • font-weight: bold • text-align: right


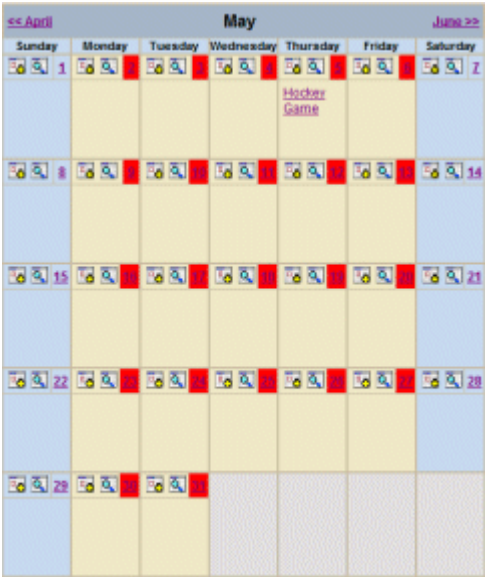
Style Class	Description	Attributes and Default Settings
.dv_DayHeader	<p>The day header for the calendar. The following example shows the day header in red.</p> 	<ul style="list-style-type: none"> • background: #99DDFF • color: #0033AA • font-family: Arial • font-size: 11pt • font-weight: bold • text-align: center
Month View (month)		

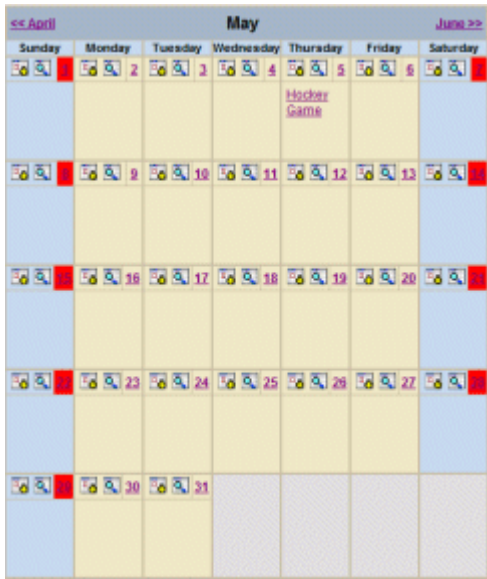

Style Class	Description	Attributes and Default Settings
.mv_weekendDOWHeader	<p>Weekend day of week header. The following example shows the weekend day of week header in red.</p> 	<ul style="list-style-type: none"> • background: #c5daef • font-family: arial • font-weight: bold • font-size: 8pt • border-top: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid • border-right: #c9bda5 1px solid • border-left: #c9bda5 1px solid • text-align: center


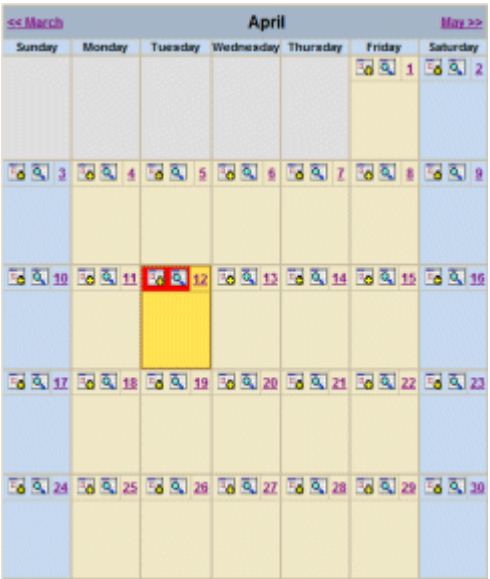
Style Class	Description	Attributes and Default Settings
.mv_weekdayDOWHeader	<p>Weekday day of week header. The following example shows the weekday day of week header in red.</p> 	<ul style="list-style-type: none"> • background: #c5daef • font-family: arial • font-weight: bold • font-size: 8pt • border-top: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid • border-right: #c9bda5 1px solid • border-left: #c9bda5 1px solid • text-align: center



Style Class	Description	Attributes and Default Settings
.mv_MonthHeader	<p>Month header. The following example shows the month header background in red.</p> 	<ul style="list-style-type: none"> • font-weight: bold • font-size: 12pt • background: transparent • font-family: Arial
.mv_MonthHeaderBkg	<p>Month header background. The following example shows the month header background in red.</p> 	<ul style="list-style-type: none"> • border-right: #c9bda5 2px solid • border-top: #c9bda5 2px solid • border-left: #c9bda5 2px solid • font-weight: bold • font-size: 9pt • background: #A5B7C9 • font-family: arial



Style Class	Description	Attributes and Default Settings
.mv_MonthHeaderNext	<p>The button for moving the calendar to the next month. The following example shows the month header next background in red.</p> 	<ul style="list-style-type: none"> • font-weight: bold • font-size: 9pt • background: transparent • font-family: Arial • text-align: right
.mv_MonthHeaderPrev	<p>The button for moving the calendar to the previous month. The following example shows the month header previous background in red.</p> 	<ul style="list-style-type: none"> • font-weight: bold • font-size: 9pt • background: transparent • font-family: Arial • text-align: left



Style Class	Description	Attributes and Default Settings
.mv_NonMonthBorder	<p>The cells at the beginning or end of the month that do not contain days. The following example shows the non month border in red.</p>  <p>The image shows a calendar for May. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday. The dates 1 through 31 are shown in a grid. The cells for the 29th, 30th, and 31st are highlighted in red, indicating they are non-month border cells.</p>	<ul style="list-style-type: none"> • border-right: #c9bda5 1px solid • border-top: #c9bda5 1px solid • background: #e0e0e0 • border-left: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid
.mv_DateCell	<p>The cells where the numerical weekdays appear. The following example shows the date cell in red.</p>  <p>The image shows a calendar for May. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday. The dates 1 through 31 are shown in a grid. The cells for the 1st, 2nd, 3rd, 4th, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, 14th, 15th, 16th, 17th, 18th, 19th, 20th, 21st, 22nd, 23rd, 24th, 25th, 26th, 27th, 28th, 29th, 30th, and 31st are highlighted in red, indicating they are date cells.</p>	<ul style="list-style-type: none"> • font-weight: bold • font-size: 9pt • background: #F0E9C5 • border-left: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid • font-family: arial


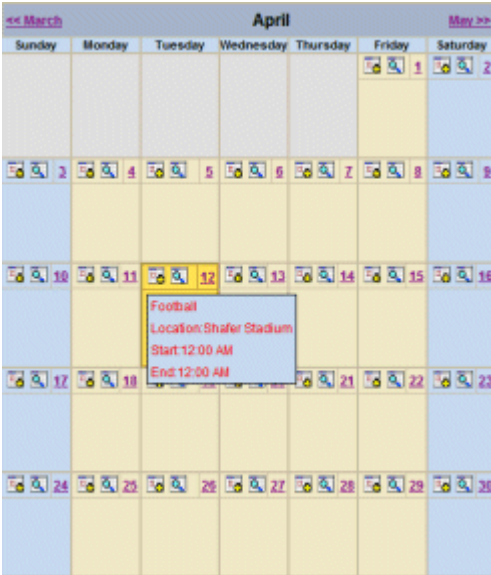
Style Class	Description	Attributes and Default Settings
.mv_DateCellWeekend	<p>The cells where the numerical weekend days appear. The following example shows the date cell weekend in red.</p>  <p>The screenshot shows a calendar for the month of May. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The dates are displayed in a grid. The weekends (Sundays and Saturdays) are highlighted in red. The date 4 is highlighted with a 'Hockey Game' event. The date 31 is highlighted in red.</p>	<ul style="list-style-type: none"> • font-weight: bold • font-size: 9pt • background: #C5DAEF • border-left: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid • font-family: arial
.mv_DateCellToday	<p>The cells where today's numerical day appears. The following example shows the date cell today in red.</p>  <p>The screenshot shows a calendar for the month of April. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The dates are displayed in a grid. The date 11 is highlighted in red, indicating it is the current day. The date 1 is highlighted with a 'Hockey Game' event. The date 31 is highlighted in red.</p>	<ul style="list-style-type: none"> • background: #FFE354 • border-left: #b38b3b 1px solid • border-bottom: #b38b3b 1px solid • font-family: Arial • font-size: 9pt • font-weight: bold


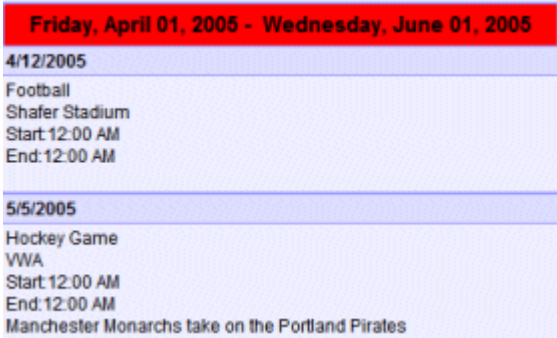
Style Class	Description	Attributes and Default Settings
<p><code>.mv_DateCellEmptyWeekend</code></p> <p>Note: These cells appear empty when you are not logged in to Ektron CMS400.NET.</p>	<p>The weekend cells that hold the Add Event and View Event buttons. The following example shows the date cell empty weekend in red.</p>  <p>The screenshot shows a calendar for the month of May. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The dates are arranged in a grid. The weekends (Sundays and Saturdays) are highlighted with a red background. The text 'Hockey Game' is visible on Thursday, May 5th.</p>	<ul style="list-style-type: none"> ● background: #C5DAEF ● border-bottom: #c9bda5 1px solid ● font-family: Arial ● font-size: 9pt ● font-weight: bold
<p><code>.mv_DateCellEmptyToday</code></p> <p>Note: This cell appears empty when you are not logged in to Ektron CMS400.NET.</p>	<p>The today cell that holds the Add Event and View Event buttons. The following example shows the date cell empty today in red.</p>  <p>The screenshot shows a calendar for the month of April. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The dates are arranged in a grid. The date April 12th is highlighted with a red background. The text 'Hockey Game' is visible on Friday, April 15th.</p>	<ul style="list-style-type: none"> ● background: #FFE354 ● border-bottom: #B38B3B 1px solid ● font-family: Arial ● font-size: 9pt ● font-weight: bold

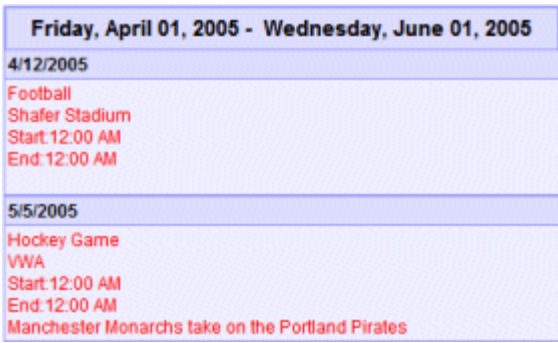
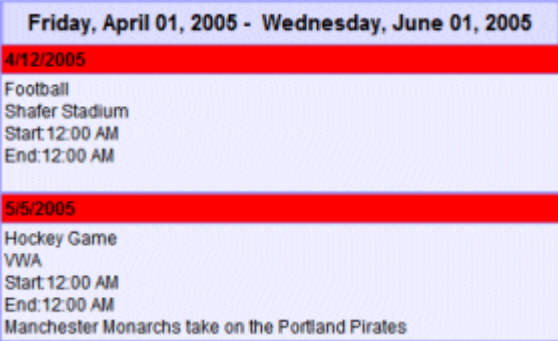
Style Class	Description	Attributes and Default Settings
<p><code>.mv_DateCellEmpty</code></p> <p>Note: These cells appear empty when you are not logged in to Ektron CMS400.NET</p>	<p>The weekday cells that holds the Add Event and View Event buttons. The following example shows the date cell empty in red.</p> 	<ul style="list-style-type: none"> • background: #f0e9c5 • border-bottom: #c9bda5 1px solid • font-family: Arial • font-size: 9pt • font-weight: bold
<p><code>.mv_eventContainer</code></p>	<p>The popup window that appears when the event is hovered over. The following example shows the event container circled in red.</p> 	<ul style="list-style-type: none"> • visibility: hidden • position: absolute <p>Note: When the visibility property is set to visible, the event container appears upon page load.</p>

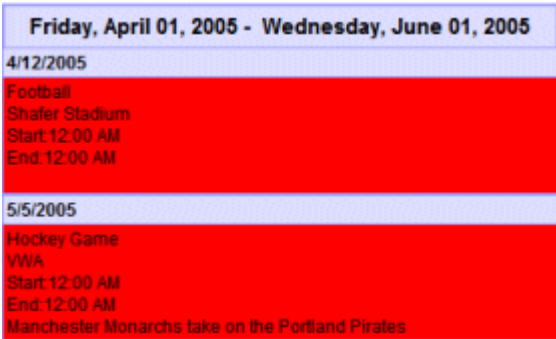
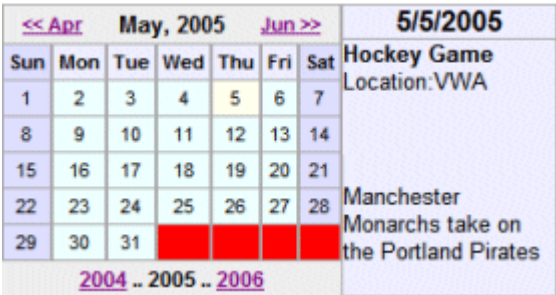
Style Class	Description	Attributes and Default Settings
.mv_dayBorderToday	<p>The today cell that holds listed events. The following example shows the day border today in red.</p> 	<ul style="list-style-type: none"> • border-right: #b38b3b 2px solid • border-top: #b38b3b 2px solid • background: #ffe354 • border-left: #b38b3b 2px solid • border-bottom: #b38b3b 2px solid
<p>.mv_dayBorder</p> <p><u>This class effects all weekday cells, except the today cell.</u></p>	<p>The weekday cell that holds listed events. The following example shows the day border in red.</p> 	<ul style="list-style-type: none"> • border-right: #c9bda5 1px solid • border-top: #c9bda5 1px solid • background: #f0e9c5 • border-left: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid

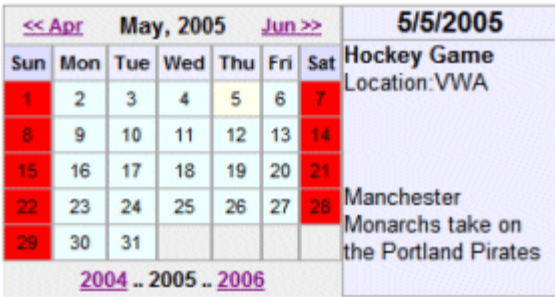
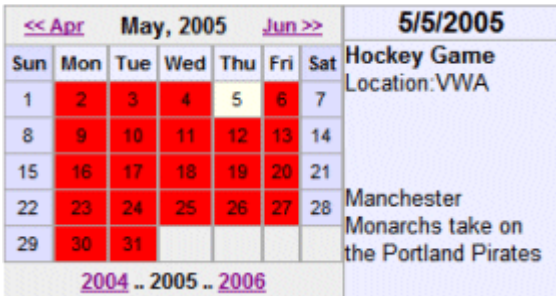
Style Class	Description	Attributes and Default Settings
.mv_dayBorderWeekend	<p>The Weekend day cell that holds listed events. The following example shows the day border weekend in red.</p> 	<ul style="list-style-type: none"> • border-right: #c9bda5 1px solid • border-top: #c9bda5 1px solid • background: c5daef • border-left: #c9bda5 1px solid • border-bottom: #c9bda5 1px solid
.mv_eventTitle	<p>The event title cell. The following example shows the event title cell in red.</p> 	<ul style="list-style-type: none"> • font-weight: normal • font-size: 9pt • background: transparent • font-family: Arial

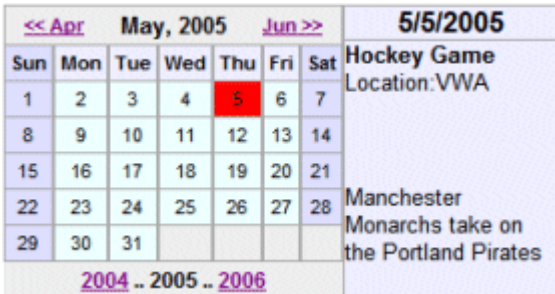
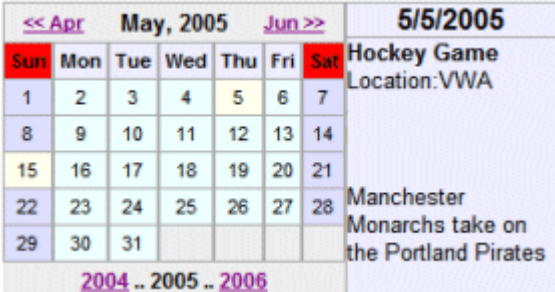
Style Class	Description	Attributes and Default Settings
.mv_DayCellToday	<p>Today's event title cell. The following example shows the day cell today in red.</p> 	<ul style="list-style-type: none"> • background: transparent • font-family: Arial • font-weight: bold • font-size: 9pt
.mv_TodayCell	<p>The font for the event pop-up. The following example shows the today cell font in red.</p> 	<ul style="list-style-type: none"> • font-weight: normal • font-size: 9pt • background: #C5DAEF • font-family: Arial

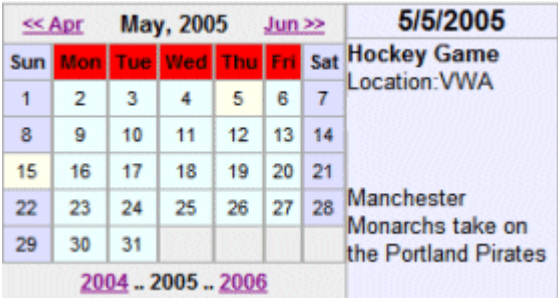
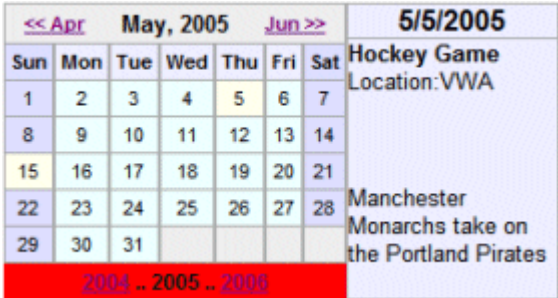
Style Class	Description	Attributes and Default Settings
.mv_eventPopup	<p>The border for the event pop-up. The following example shows the event pop-up border in red.</p> 	<ul style="list-style-type: none"> • border-right: #000000 1px solid • border-top: #000000 1px solid • border-left: #000000 1px solid • border-bottom: #000000 1px solid
Inline View (inline)		
.ILViewHeaderBkg	<p>The inline view header background. The following example shows the view header background in red.</p> 	<ul style="list-style-type: none"> • border-right: #9999ff 2px solid • border-top: #9999ff 2px solid • font-weight: bold • font-size: 11pt • background: #ddddff • border-left: #9999ff 2px solid • font-family: Arial • text-align: center

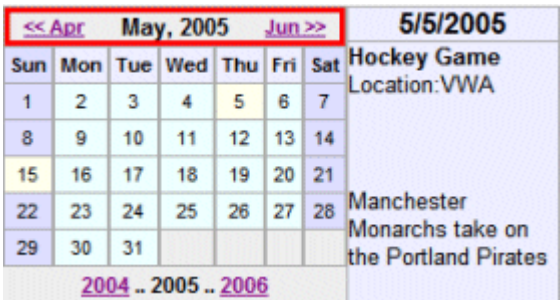
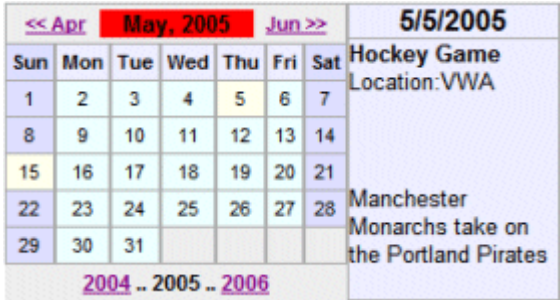
Style Class	Description	Attributes and Default Settings
.ILViewEvent	<p>Font for the event description. The following example shows the view event font in red.</p> 	<ul style="list-style-type: none"> • font-size: 9pt • font-family: Arial
.ILViewDate	<p>The date for the inline view. The following example shows the view date background in red.</p> 	<ul style="list-style-type: none"> • border-right: #9999ff 1px solid • border-top: #9999ff 1px solid • font-weight: bold • font-size: 9pt • background: #ddddff • border-left: #9999ff 1px solid • font-family: Arial

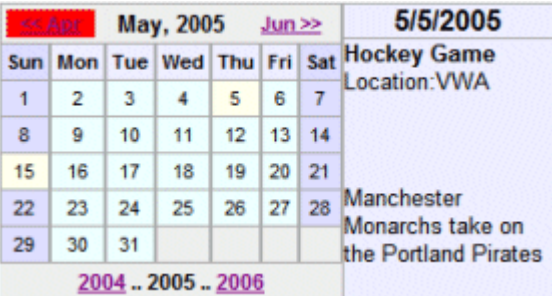
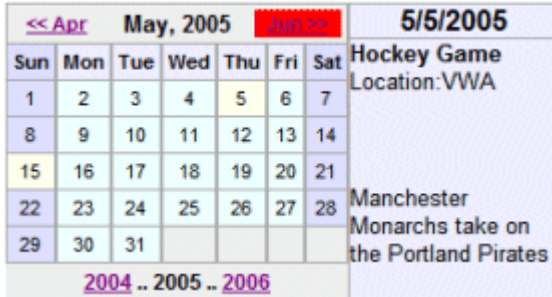

Style Class	Description	Attributes and Default Settings
.ILViewDayCell	<p>The cell the events are listed in. The following example shows the view day cell in red.</p> 	<ul style="list-style-type: none"> • border-right: #9999ff 1px solid • border-top: #9999ff 1px solid • font-weight: bold • font-size: 9pt • background: #eeeeff • border-left: #9999ff 1px solid • border-bottom: #9999ff 1px solid • font-family: Arial
Month-Up with Event (monthupwithevent)		
.mup_nonMonthCell	<p>The cells at the beginning or end of the month that do not contain days. The following example shows the non-month cells in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: normal • font-size: 8pt • text-align: center • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid

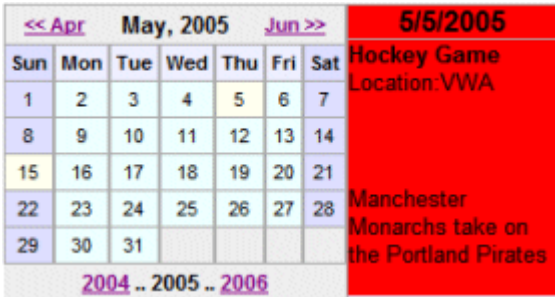
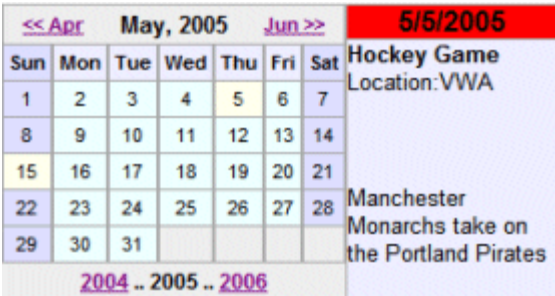
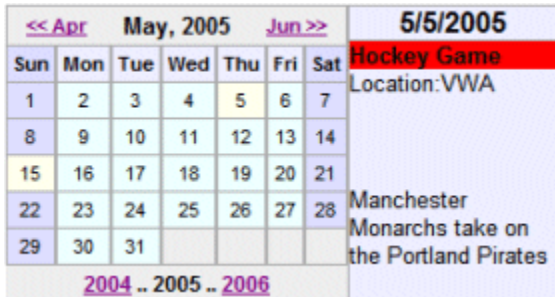
Style Class	Description	Attributes and Default Settings
.mup_weekendCell	<p>The weekend day cell. The following example shows the weekend cell in red.</p> 	<ul style="list-style-type: none"> • background: #dedeff • font-family: Arial • font-weight: normal • font-size: 8pt • text-align: center • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid
.mup_weekdayCell	<p>The week day cell. The following example shows the week day cell in red.</p> 	<ul style="list-style-type: none"> • background: #eeffff • font-family: Arial • font-weight: normal • font-size: 8pt • text-align: center • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid

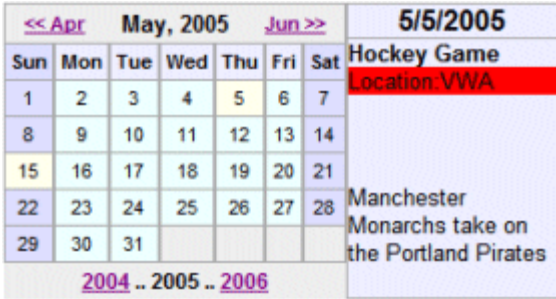
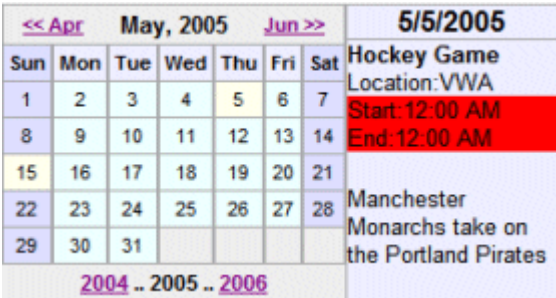
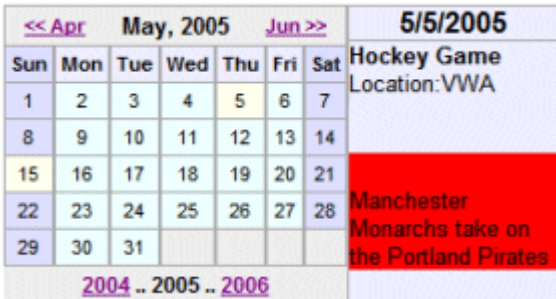
Style Class	Description	Attributes and Default Settings
.mup_weekdayEventCell	<p>Highlights the weekdays that have events. The following example shows the weekday event cell in red.</p> 	<ul style="list-style-type: none"> • background: #ffffee • font-family: Arial • font-weight: normal • font-size: 8pt • text-align: center • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid • cursor: pointer
.mup_weekendDOWHeader	<p>Weekend day of week header. The following example shows the weekend day of week header in red.</p> 	<ul style="list-style-type: none"> • background: #dddfff • font-family: Arial • font-weight: Bold • font-size: 8pt • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid • text-align: center



Style Class	Description	Attributes and Default Settings
.mup_weekdayDOWHeader	<p>Weekday day of week header. The following example shows the weekday day of week header in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeff • font-family: Arial • font-weight: Bold • font-size: 8pt • border-right: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid • text-align: center
.mup_yearFooter	<p>The year selection cell. The following example shows the year footer in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: Bold • font-size: 9pt • text-align: center



Style Class	Description	Attributes and Default Settings
.mup_monthHeaderBorder	<p>The border around the month header. The following example shows the month header border in red.</p>  <p>The screenshot shows a month header for May 2005. The navigation area (including '<< Apr', 'May, 2005', 'Jun >>', and the date '5/5/2005') is enclosed in a red border. Below this is a calendar grid for May, and to the right, text about a 'Hockey Game' at 'VWA' and 'Manchester Monarchs take on the Portland Pirates'. At the bottom, there is a year navigation bar '2004 .. 2005 .. 2006'.</p>	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: Bold • font-size: 10pt • border-right: #aaaaaa 1px solid • border-top: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid • text-align: center
.mup_monthHeader	<p>The month header. The following example shows the month header in red.</p>  <p>This screenshot is identical to the one above, but the month name 'May' in the navigation bar is highlighted in red, while the rest of the header area is not bordered.</p>	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: Bold • font-size: 10pt • text-align: center

Style Class	Description	Attributes and Default Settings
.mup_monthHeaderPrev	<p>The previous button, on the month header. The following example shows the month header previous back ground in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: Bold • font-size: 8pt • text-align: center
.mup_monthHeaderNext	<p>The next button, on the month header. The following example shows the month header next background in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeee • font-family: Arial • font-weight: Bold • font-size: 8pt • text-align: center
.mup_eventDispBkg <u>Note: .mup_eventDispBkg deals with the background layer of the event display. .mup_eventDisplay deals with the layer the event is displayed on.</u>	<p>The background for the event display. The following example shows the event display background in red.</p> 	<ul style="list-style-type: none"> • background: #eeeeff • border-right: #aaaaaa 1px solid • border-top: #aaaaaa 1px solid • border-bottom: #aaaaaa 1px solid • border-left: #aaaaaa 1px solid

Style Class	Description	Attributes and Default Settings
<p><code>.mup_eventDisplay</code></p> <p><u>Note: <code>.mup_eventDispBkg</code> deals with the background layer of the event display. <code>.mup_eventDisplay</code> deals with the layer the event is displayed on.</u></p>	<p>The event display. The following example shows the event display in red.</p> 	<p>background: transparent</p>
<p><code>.mup_eventDisplayDate</code></p>	<p>The event display date header. The following example shows the event display date in red.</p> 	<ul style="list-style-type: none"> • font-size: 12pt • font-weight: bold • background: transparent • font-family: Arial • text-align: center • border-bottom: #aaaaaa 1px solid
<p><code>.mup_eventDisplayEventTitle</code></p>	<p>Display for the event title. The following example shows the event display event title in red.</p> 	<ul style="list-style-type: none"> • font-weight: bold • background: transparent • font-family: Arial • font-size: 10pt • text-align: left

Style Class	Description	Attributes and Default Settings
.mup_eventDisplayLocation	<p>Display for the event location. The following example shows the event display location in red.</p> 	<ul style="list-style-type: none"> • background: transparent • font-family: Arial • font-weight: normal • font-size: 10pt • text-align: left
.mup_eventDisplayTimes	<p>Display for event times. The following example shows the event display times in red.</p> 	<ul style="list-style-type: none"> • background: transparent • font-family: Arial • font-weight: normal • font-size: 10pt • text-align: left
.mup_eventDisplayLongDesc	<p>Displays the event's long description. The following example shows the event display long description in red.</p> 	<ul style="list-style-type: none"> • background: transparent • font-family: Arial • font-weight: normal • font-size: 10pt • text-align: left
Event Type Control (used in month display)		

Style Class	Description	Attributes and Default Settings
.etCtrl_Background	<p>The background of the event type control cell. The following example shows the event type control background in red.</p> 	<ul style="list-style-type: none"> • border-right: #c9bda5 2px solid • border-top: #c9bda5 2px solid • border-left: #c9bda5 2px solid • border-bottom: #c9bda5 2px solid • font-size: 9pt • background: #A5B7C9 • font-family: Arial
.etCtrl_InstructCell	<p>The instructions cell for event type control. The following example shows the instruction cell in red.</p> 	<ul style="list-style-type: none"> • background: transparent • font-family: Arial • font-size: 9pt • font-weight: bold • text-align: right • vertical-align: top

Style Class	Description	Attributes and Default Settings
.etCtrl_SelCell	<p>The cell the selection box is displayed in. The instructions cell for event type control. The following example shows the selection cell in red.</p> 	<ul style="list-style-type: none"> ● background: transparent ● font-family: Arial ● font-size: 9pt ● text-align: left ● vertical-align: top
.etCtrl_selectBox	<p>The event type selection box. The instructions cell for event type control. The following example shows the selection box in red.</p> 	<ul style="list-style-type: none"> ● background: #c5daef

designformentry.css

Used internally by the Ektron CMS400.NET. It should not be changed.

pop_style.css

This file is located in `webroot/CMS400Demo/Workarea/csslib`. It is used to customize the look of the DhtmlMenu server control. The following is a list of style sheet properties that can be modified.

Style Class	Description	Attribute	Default Value
.lowText	Text of the floating menu as displayed on the menu.	<ul style="list-style-type: none">• color• font• text-align• text-decoration	<ul style="list-style-type: none">• #000000 (Black)• 12px Arial, sans-serif• left• none
.highText	Text of the floating menu when highlighted.	<ul style="list-style-type: none">• color• font• text-align• text-decoration	<ul style="list-style-type: none">• #000000 (Black)• 12px Arial, sans-serif• left• none
.roothighText <u>Note: This class only works when called by the CssClass property. Use the Font property in the Appearance category to define root menu text.</u>	Text of the root menu as displayed on a Web page.	<ul style="list-style-type: none">• color• font• text-align• text-decoration	<ul style="list-style-type: none">• #000000 (Black)• 16px Arial, sans-serif• center• none
.itemBorder	Border color and size of the menu.	<ul style="list-style-type: none">• border	<ul style="list-style-type: none">• 1px solid #666666
.itemBorderBlank	Separator border color and size of the menu.	<ul style="list-style-type: none">• border	<ul style="list-style-type: none">• 1px solid #CCCCDD

CMS400Demo.css

This file is located in `webroot/CMS400Demo`. It is used to customize the look of the Ektron Medical Demo site.

Developer Styles.css

The `Styles.css` file is located in `webroot/CMS400Demo/Developer`. It is a general style sheet provided for creating a new Web application project.

Default.css

This file is located in `webroot/CMS400Demo/Developer`. It is the default CSS file for the Developer Web site ASPX files.

List Summary OrderedList.css

The `OrderedList.css` file is located in `webroot/CMS400Demo/Developer/ListSummary`. It is used to format the `ListSummaryOrderedList.aspx` page.

List Summary unorderedlist.css

The `unorderedlist.css` file is located in `webroot/CMS400Demo/Developer/ListSummary`. It is used to format an unordered List Summary.

Collection OrderedList.css

The `OrderedList.css` file is located in `webroot/CMS400Demo/Developer/Collection`. It is used to format the `CollectionOrderedList.aspx` page.

Collection unorderedlist.css

The `unorderedlist.css` file is located in `webroot/CMS400Demo/Developer/Collection`. It is used to format an unordered Collection.

CSSMenu.css

This file is located in `webroot/CMS400Demo/Developer/Menu`. It is used to format the `cssmenu.aspx` page.

Search unorderedList.css

The `unorderedlist.css` file is located in `webroot/CMS400Demo/Developer/Search`. It is used to format the `SearchOrderedList.aspx` page.

Discussion Board CSS Files

The following files are used to customize the look of the Discussion Board. These files are located in `webroot/CMS400Demo/Workarea/threadeddisc/themes`.

-
- Chrome.css
 - cool.css
 - jungle.css
 - modern.css
 - royal.css
 - slate.css
 - standard.css
 - techno.css
 - theme.css

Additional Information

Supporting Documents

You can find more detailed information about Ektron CMS400.NET in the following manuals:

- Ektron CMS400.NET Administrator Manual
- Ektron CMS400.NET User Manual
- Ektron CMS400.NET Quick Start Manual
- Ektron CMS400.NET Setup Manual

Support

If you are experiencing trouble with any aspect of Ektron CMS400.NET, please contact the Ektron Support Team via our Web Site at

http://www.ektron.com/support/cms400_support.aspx

Sales

For questions and answers about purchasing Ektron CMS400.NET from Ektron, contact the sales department by sending an e-mail to:

sales@ektron.com

Information

For general information about Ektron and its products, you may send an e-mail to:

info@ektron.com

Ektron, Inc.
5 Northern Blvd., Suite 6
Amherst, NH 03031
Email: sales@ektron.com
Tel: +1 603-594-0249
Fax: +1 603-594-0258